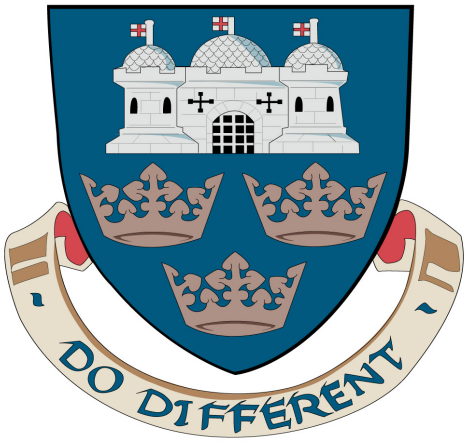
University of East Anglia

Acoustic Approaches to Gender and Accent Identification

by
Andrea DeMarco

A thesis submitted for the Degree of
Doctor of Philosophy

School of Computing Sciences
University of East Anglia, England



Supervisor: Prof. Stephen J. Cox
Co-supervisor: Dr. Ben Milner

June 2015

Abstract

There has been considerable research on the problems of speaker and language recognition from samples of speech. A less researched problem is that of accent recognition. Although this is a similar problem to language identification, different accents of a language exhibit more fine-grained differences between classes than languages. This presents a tougher problem for traditional classification techniques. In this thesis, we propose and evaluate a number of techniques for gender and accent classification. These techniques are novel modifications and extensions to state of the art algorithms, and they result in enhanced performance on gender and accent recognition.

The first part of the thesis focuses on the problem of gender identification, and presents a technique that gives improved performance in situations where training and test conditions are mismatched.

The bulk of this thesis is concerned with the application of the i-Vector technique to accent identification, which is the most successful approach to acoustic classification to have emerged in recent years. We show that it is possible to achieve high accuracy accent identification without reliance on transcriptions and without utilising phoneme recognition algorithms. The thesis describes various stages in the development of i-Vector based accent classification that improve the standard approaches usually applied for speaker or language identification, which are insufficient. We demonstrate that very good accent identification performance is possible with acoustic methods by considering different i-Vector projections, frontend parameters, i-Vector configuration parameters, and an optimised fusion of the resulting i-Vector classifiers we can obtain from the same data.

We claim to have achieved the best accent identification performance on the test corpus for acoustic methods, with up to 90% identification rate. This performance is even better than previously reported acoustic-phonotactic based systems on the same corpus, and is very close to performance obtained via transcription based accent identification. Finally, we demonstrate that the utilization of our techniques for speech recognition purposes leads to considerably lower word error rates.

Keywords: Accent Identification, Gender Identification, Speaker Identification, Gaussian Mixture Model, Support Vector Machine, i-Vector, Factor Analysis, Feature Extraction, British English, Prosody, Speech Recognition.

Acknowledgements

I would like to thank all the people who supported me during the realization of this thesis. In particular, I would like to thank Prof. Stephen Cox, who supervised this work, for his excellent guidance and friendship. The countless times we have sat down together to break through the many barriers in the last four years have propelled me forward to complete this work. It seems not too long ago when I was first formulating my research proposal, I realised that the supervision I would get would be a perfect match for me. This has been a constant throughout. Thanks also goes to Dr. Ben Milner, the co-supervisor for this thesis. I would also like to thank the University of East Anglia for providing me with a scholarship to undertake this research, and my viva examiners, Dr. Gavin Cawley and Dr. Tomi Kinnunen.

None of this would have been possible without the affectionate and constant support of my family, who have had confidence in me all throughout, serving as reassurance in my abilities. I am also very grateful to be surrounded by good friends who have provided moral support during this time. For all the hard work and stamina one can have, it is equally important to know when and how to wind down and rest. My apologies go to all those who have endured my whining and dramatic ups and downs.

Another big thanks goes to my laboratory colleagues (Dominic Howell, Jason Lines, John Taylor, Philip Harding, Helen Bear, Faheem Khan and Sarah Taylor) at the Audio, Speech and Language Processing Laboratory of the University of East Anglia. We have all shared this experience together, and we have all learnt a lot more than we could have done on our own. Thanks goes to my other colleagues at the School of Electronic, Electrical and Computer Engineering of the University of Birmingham, in particular Prof. Martin Russell and Maryam Najafian for the exciting work we did together over the last year. Thanks also to the various colleagues and peers I have met in various conferences during my research. It was great fun discussing ideas with you — somewhat inferring that the content of my research can be particularly interesting to some.

The research presented in this thesis was carried out on the High Performance Computing Cluster supported by the Research and Specialist Computing Support service at the University of East Anglia, and I thank the team for their support during the last four years.

With and through all of you, I am asymptotically one step closer to understanding what science is. I have come to know why I made the choice to undertake this research years ago — a choice I may not have fully understood then.

Dedicated to my family
and friends

Contents

List of Figures

List of Tables

List of Abbreviations

AID	**A**ccent **Id**entification
AMDF	**A**verage **M**agnitude **D**ifference **F**unction
ASR	**A**utomatic **S**peech **R**ecognition
CDF	**C**umulative **D**istribution **F**unction
CMN	**C**epstral **M**ean **N**ormalization
DFT	**D**iscrete **F**ourier **T**ransform
FFT	**F**ast **F**ourier **T**ransform
GA	**G**enetic **A**lgorithm
GID	**G**ender **Id**enficiation
GLDS	**G**eneralized **L**inear **D**iscriminant **S**equence
GMM	**G**aussian **M**ixture **M**odel
HMM	**H**idden **M**arkov **M**odel
IDF	**I**nverse **D**ocument **F**requency
ISC	**I**nter-**S**ession **C**ompensation
JFA	**J**oint **F**actor **A**nalysis
LDA	**L**inear **D**iscriminant **A**nalysis
LID	**L**anguage **Id**entification
LLR	**L**og **L**ikelihood **R**atio
LM	**L**anguage **M**odel
MFCC	**M**el **F**requency **C**epstral **C**oefficients
MVN	**M**ean and **V**ariance **N**ormalization
NAP	**N**uisance **A**trribute **P**rojection
PCA	**P**rincipal **C**omponent **A**nalysis
PMC	**P**arallel **M**odel **C**ombination

PPRLM	**P**arallel **P**hone **R**ecognition **L**anguage **M**odel
PRLM	**P**hone **R**ecognition **L**anguage **M**odel
QDA	**Q**udaratic **D**iscriminant **A**nalysis
RP	**R**eceived **P**ronunciation
SDC	**S**hifted **D**elta **C**epstra
SID	**S**peaker **Id**entification
SVD	**S**ingular **V**alue **D**ecomposition
SVM	**S**upport **V**ector **M**achine
SVM	**S**upport **V**ector **M**achine
VQ	**V**extor **Q**uantization
VTE	**V**erbal **T**ransformation **E**ffects
VTLN	**V**ocal **T**ract **L**ength **N**ormalisation
WCCN	**W**ithin **C**lass **C**ovariance **N**ormalization

Chapter 1

Introduction

This chapter will give an overview of the research presented in this thesis. Particularly we will look at the research questions that are explored, the reasons why they should be addressed, and the general context in which they are addressed. Finally, we also give a breakdown of the rest of the chapters in this thesis.

1.1 Research Aim

The speech signal carries a wealth of information about the speaker. The goal of speech from a linguistic perspective, is to convey a message from a speaker to a listener. Speech to text transcription is the goal of Automatic Speech Recognition (ASR). But this information is encoded differently according to the language, dialect and accent being utilized. From an acoustic perspective, there are individual characteristics that define the voice of a speaker, or a speaking style, the approximate age of a speaker, the mood the speaker is in, and in some cases, speech pathologies when there is something evidently wrong with the voice production mechanism.

This range of information makes the study of voice very multidisciplinary, and the computational methods that can be devised to model, characterise, and study speech data have different uses, not just in computing science, but in voice forensics, medical analysis, speech therapy and linguistic studies, to name a few. It is important to discover and improve ways to measure, model and compare speech characteristics as accurately and as robustly as possible.

The idea of inferring the speaker's identity from speech has many practical applications. This task is generally split up into two sub-problems. The first is identification, where given

a speech sample from a speaker, we would like to identify the person from among a pool of possible identities. The second is verification, where we have a prior claim of who the speaker is, and given a voice sample from the claimant, we want to ascertain that this is either true or false. These two general themes of speaker identification and verification can be extended to other speaker characteristics such as age, gender, language, accent etc.

Being able to have a complete speaker profile based on all these categories is desirable. Any security system based on this form of biometric identification is prone to spoofing, so having a multi-dimensional speaker profile would make it even more secure e.g. if it were possible to successfully mimic a speaker's overall acoustic characteristics, but not their accent, then the user is locked out. But security is not the only application area for such a technology. Manually labelling corpora for future analysis is very costly and time-consuming. With the onset of "big data" and the voluminous amount of (cheap) unlabelled multimedia data available, systems based on identifying speaker characteristics such as language, accent, gender, age etc. can be used for unsupervised annotation, provided they are reliable enough.

Speech classification problems are often split into two methods of analysis: supervised and unsupervised methods. In supervised analysis, the transcription of the speech at some level and at some accuracy, is known prior to acoustic analysis. If we know the transcription of an utterance, then we can devise a model that is context-based, bound by the transcription. Measurements and comparisons are then based on template models for that linguistic content. Unsupervised methods are used when there is no transcription available. The expectation (and reality) is that supervised systems perform better than unsupervised methods, but the application of unsupervised methods is much broader.

1.2 Defining Accents

There are generally a number of frequently conflated definitions of what an accent is. We choose a definition that is useful for this thesis. In Volume I of "Accents of English", the definition Wells gives is "a pattern of pronunciation used by a speaker for whom English is the native language or, more generally, by the community or social grouping to which he or she belongs" [5]. This thesis therefore will not consider dialect, which is about the use of specific words or phrases that are specific to one region but not another. For example, when a speaker of US English uses the words "elevator" and "sidewalk" to refer to what a British English speaker means by "lift" and "pavement", this is an example of *dialect*. However, when somebody from the north of England

pronounces “bath” with the same vowel sound as “rat”, rather than the vowel in “part”, as a southern English speaker would, this is an example of *accent*.

In the UK, Received Pronunciation (RP) is usually the accent generally used by national news broadcasters (although this is not always the case). This does not mean that there is a correct or incorrect way of pronouncing English, but it does imply that the articulatory phonetics differ from one accent to another [6].

Furthermore, this thesis is concerned with the accents of native British English speakers from the British Isles. One can categorize British English accents into five broad regions: the North and South of England, Scotland, Wales and Ireland. The northern region can be split further into the Midlands, mid-North and far-North. The southern region can be split further into the London area, the surrounding counties, East Anglia and the South-West [7]. However, splitting accents into groups does not mean that each accent group has a single particular set of characteristics unique to itself. There may well be variability of the same accent within a particular accent group, as well as some shared traits across multiple accent groups. Accents should therefore be more precisely defined as existing within a continuous space representation.

Over recent years, accents have been receiving more attention in speech processing, but the attention to classification has been relatively poor. For instance, in Interspeech 2013 (the major conference for speech processing), there were a total of eleven papers related to accent and language identification. Only three of those were about accents as we define it above. The others were in the more popular area of language classification. But since accents are a major source of variation in speech, together with gender and speaker differences, having good accent identification performance has potential applications in ASR, annotation, biometric profiling etc. The recent work in accent identification has followed from the state of the art in language identification. Our work primarily follows from the recent work on accent classification by Hanani et al. [4], which used the same accent dataset as the one we utilize in our research.

The current state of the art Speaker Identification(SID)/Accent Identification(AID)/Language Identification(LID) is based on short-term acoustic features that are variants of Mel Frequency Cepstral Coefficients (MFCC). These features describe the short-term spectral characteristics of speech. The basic idea behind modelling these features via adapted Gaussian Mixture Models (GMM) is that when class specific features are chosen, say for speakers in SID, or languages in LID, or accents in AID, the resulting model would generalize well towards the class it has to represent. Also, the short-term duration of speech represents pseudo-phones or pseudo-syllables — and it is a segmentation that is unsupervised, which is important for

acoustic-only classification methods.

However, it will become very apparent that there the major source of error in AID is the inherent acoustic difference in speakers within or across the same accent group. This, coupled with the unsupervised nature of our analysis, makes it very difficult for a GMM to accurately model the class. The accent information is relatively diluted by these speaker differences, as well as by generic modelling of phonetic inventory. We will therefore demonstrate the extent of how the standard statistical modelling technique of adapted Gaussian Mixture Models, whilst being quite appropriate for tasks such as SID, is inadequate for the AID task. For this reason, this thesis has a major focus on AID experimentation and analysis. Because of the fine-grained differences in accents as opposed to languages, it may seem surprising to some how important speaker compensation methods based on i-Vectors are crucial to this problem, especially on the acoustic-only setting.

1.3 Research Questions

This thesis is entitled "Accent and Gender Classification Based on Acoustic-Only Features". Below are the most important research questions we address:

1. Gender is one of the primary sources of speaker variation in speech. Gender Identification (GID) has been given some attention in the past. Are there any additions to standard algorithms that can improve GID performance across multiple corpora, and different speaker populations, under shifted datasets (e.g. different recording conditions) for training and testing?

2. The most successful approaches to AID have analysed the differences in phonetic realization of equivalent words or phrases spoken across different accents. However, we want to perform AID for unlabelled, text-independent modelling and classification. Can reliable AID accuracy be achieved in these conditions? How does it compare to more traditional methods of AID?

3. If traditional methods of acoustic classification do not work so well for the AID problem, what changes (if any) have to me made to the classification framework? And what is the acoustic-phonetic underpinning for these changes?

4. Are traditional acoustic-only front-end systems suitable for the AID problem? How can they be improved? Is there a difference between utilizing short-term or long-term features

within the context of acoustic-only classification?

Some of these questions will be answered in a self-contained manner. Others are answered as this thesis progresses across multiple chapters.

1.4 Chapter Breakdown

The chapters that follow are organised as follows:

Chapter 2 is a reference for the technical background for this thesis. It contains an overview of feature extraction from speech. We also describe the important acoustic-only classification techniques employed in the field of SID, LID, and AID. In particular we focus on the GMM, Support Vector Machines (SVM) and some commonly used dimensionality reduction techniques.

Chapter 3 gives an overview of the literature on GID and AID systems, whilst touching on SID systems where relevant. It will focus on identification techniques as a whole, and will include material that is not intended solely for acoustic-only AID. We also give an overview on human and animal perception of speech and acoustics. The chapter ends with considerable detail on the i-Vector paradigm and how it evolved from the standard GMM paradigm.

Chapter 4 will describe the datasets used in the various experiments performed in this thesis. In particular we shall describe the TIMIT, WSJCAM0, and ABI-1 corpora. The TIMIT, WSJCAM0 and ABI-1 datasets are used for our experiments in Chapter 5. The ABI-1 is used exclusively in Chapters 6, 7 and 8.

Chapter 5 presents our research in the area of GID. It will evaluate the performance of standard techniques on GID, and show some of the weaknesses of these systems. It then proposes modifications to the standard algorithm based on pitch models for specific acoustic contexts, and the agreements between classifiers when pitch shifting is applied to the original signal. The results are compared to show that our proposed modifications are robust to changes in speaker sets and corpora.

Chapter 6 presents work on AID using short-term feature vectors. We investigate the differences across different classification techniques, prior to the introduction of the i-Vector framework. We also evaluate the utility of some long-term prosodic features on AID. This chapter will evaluate the difficulty in unsupervised/unlabelled prosodic features for AID over a number of different experiments.

Chapter 7 presents our work on AID with the application of the state-of-the-art i-Vector paradigm in SID and LID. We show how the gains observed in SID and LID can also apply to native AID. We then discuss our proposed enhancements to the scoring mechanism used by the classifier to build what we believe to be the best unsupervised AID classifier to date. The enhancements included an iterative variant of LDA scoring, as well as a fusion mechanism that combines different i-Vector extractors together.

Chapter 8 extends the previous chapter by looking into more detail at the short-term feature vectors that are suitable for AID. We show that the ones traditionally used in SID and LID, though helpful, are generally not the best for AID. We experimentally derive an improved set of feature vectors based on different front-end configurations. We also investigate the performance of our AID system based on utterance of various durations. Furthermore, we take an initial look at how the unsupervised AID developed in this thesis can have a very practicable effect on the design of ASR systems.

Chapter 9 concludes this thesis by highlighting both the quantitative achievements, and by summarizing the salient qualitative contributions which will be useful for future work in this field.

1.5 Research Contributions

This thesis produces a number of contributions to the field of automatic classification of speaker gender and accent. They can be summarised as follows:

1. The modification of a standard GID classifier to better handle changes across different corpora, the use of context dependent pitch GMMs and the use of pitch-shifting to sort out ambiguous cases where gender scoring is not all clear. Refer to publication **1** in the next section.

2. The first investigation of the effectiveness of the i-Vector technique for identifying regional accents of British English. Refer to publication **2** in the next section.

3. The demonstration of the incompleteness of learning accent factors based on a single i-Vector configuration, which are fused at score level. Refer to publication **3** in the next section.

4. Joint work with other researchers in the field to assess the validity of unsupervised AID to

provide quasi-real-time model selection for ASR. Refer to publication **4** in the next section.

5. The analysis of AID with different front-end systems, and the gains we can get by utilising a large set of weak i-Vector systems for a final classification.

1.6 List of Publications

The following is a list of publications arising out of this thesis at the time of print:

1. **A. DeMarco** and S.J. Cox, "An Accurate and Robust Gender Identification Algorithm", Proc. Interspeech 2011, Florence, Italy, 2429-2432, 2011 [8].
2. **A. DeMarco** and S.J. Cox, "Iterative Classification of Regional British Accents in i-Vector Space", Proc. MLSLP 2012, Portland, USA, 1-4, 2012 [9].
3. **A. DeMarco** and S.J. Cox, "Native Accent Classification via I-Vectors and Speaker Compensation Fusion", Proc. Interspeech 2013, Lyon, France, 1472-1476, 2013 [10].
4. M. Najafian, **A. DeMarco**, S.J. Cox and M. Russell, "Unsupervised Model Selection for Recognition of Regional Accented Speech", Proc. Interspeech 2014, Singapore, 2014 [3].

1.7 Summary

In this chapter, we have given an overview of the thematic questions and focus that this thesis investigates. We introduced the idea of accents, and the acoustic-only classifier limitations we are imposing in our investigation. A list of publications that arise out of this work is shown, together with a breakdown of the chapters that follow.

Chapter 2

Technical Background

In this chapter we give a background to some technical material that is essential to all the experimentation and processing in this thesis. Firstly, we give an overview of the front-end analysis of speech signals. Following this is an overview of acoustic-only classification methods that are generally applied to classification problems in GID, AID and SID. This chapter will serve as a summary of essential background to understand the application of feature extraction and techniques underlying the basis of classification methods as applied in later chapters. We then go over some of the most fundamental methods to build classification systems and how the dimensionality of the feature space for these models can be reduced. Finally we give a brief overview of genetic algorithms for the purposes of the requirements of this thesis. The techniques are discussed in a mostly abstract fashion. In later chapters, we will tie individual methods and features to specific methods and experiments as tried in literature, and in our own work.

2.1 Speech Production

A general assumption in this field of study is that the more we understand about the physiological process of how speech signals are produced and further on understood in the human speech perception processes, then the closer we can approximate an artificial system that can do the same job to the same extent as humans do [11]. The speech signal is the common element between the output of the speech production system and the input to the speech perception system. We usually think about speech in terms of language and grammatical constructs such as sentences, phrases and words. However these are linguistic explanatory constructs: here, we

focus on the process of production of speech and the resulting acoustic signal.

The first stage of speech production consists of the speaker formulating the message content that has to be conveyed to a listener (a mental process). This content has somehow to be translated to a code that can be pronounced by the speaker and understood by both speaker and listener. In speech production, this code is a series of neuromuscular commands that cause the vocal apparatus to move as and when appropriate, and to shape the vocal tract in such a way that a properly intonated sequence of sounds is generated and output at the lips. The end result is an acoustic signal. The neuromuscular commands are directly responsible for simultaneously controlling all aspects of motion of the articulatory muscles [12].

The acoustic speech signal is received by the listener, and the process of decoding the information in the message is called the speech perception process. The acoustic signal is processed along the basilar membrane of the inner ear, which provides spectral analysis of the incoming signal. This spectral signal is converted to activity signals on the auditory nerve with a neural transduction process. These activity signals are converted into a language code to higher levels of processing in the brain. How this is done, is not yet very much understood. Finally the meaning of the signal (as conveyed by the speaker) is achieved [12].

The speech production process provides information on not only the information content (or message), but also the speaker's voice, gender, accent, language, etc. Similarly, the speech perception process determines not only the information the speaker wanted to communicate, but also enables a listener to listen to and 'learn' a speaker's voice and the various sub-characteristics. That is why both processes must be well understood for the purpose of determining gender, accent and speaker voice by an automatic system.

Production of speech sounds is roughly based on an air source (from the lungs) that passes through the vocal folds. These folds are either held open, or vibrate. The rate of vibration of the vocal cords is determined by their size and the muscle tension placed on them. In adult males, the vocal cords are usually longer and larger than those in children, whilst adult females are intermediate. Similar to string instruments, the longer and thicker the strings, the lower is the rate of vibration, resulting in listeners hearing a lower pitch voice. The output from vibrating vocal cords is referred to as a voiced speech signal. However unvoiced speech is also possible, when the vocal folds are open, allowing air to flow from the larynx to the vocal tract.

The air flow from voiced or unvoiced speech is then modified when passing through the vocal tract. Figure 2.1 is a simplified diagram of the human speech production system. The vocal

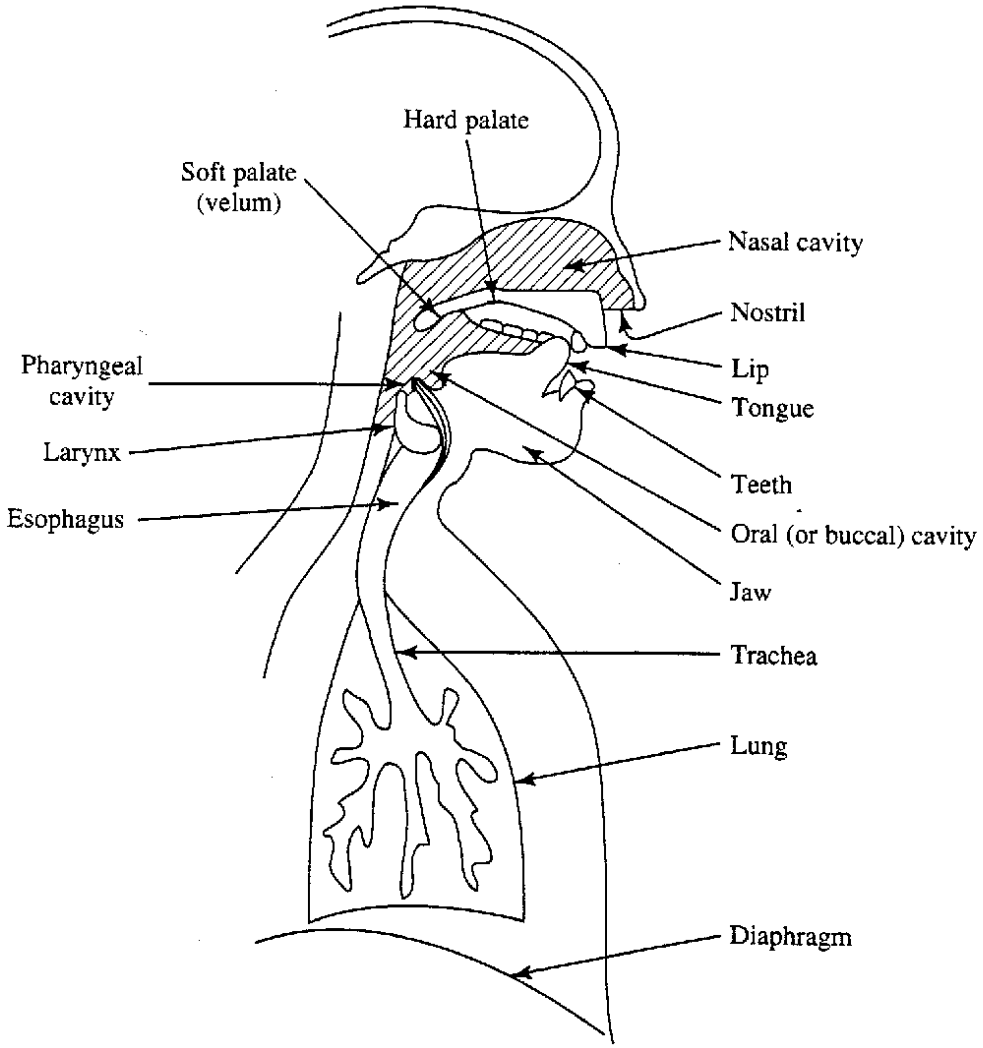


Figure 2.1: Schematic view of the human vocal mechanism [1].

tract begins at the opening of the vocal cords (found between the larynx and the esophagus), and ends at the lips. The vocal tract consists of the pharynx (connecting the esophagus to the mouth) and the mouth, or oral cavity. In the average male, the total length of the vocal tract is about 17cm. The cross-sectional area of the vocal tract, determined by the positions of the tongue, lips, jaw, and velum, varies from zero (complete closure) to about 20cm^2. The nasal tract begins at the velum and ends at the nostrils. When the velum is lowered the nasal tract joins the vocal tract to produce the nasal sounds of speech. Depending on the position of the articulators (i.e. jaw, tongue, velum, lips, mouth), different sounds can be vocalised [12].

The characteristic of the net effect of air flow that is modified as it passes through through the vocal tract is modelled, rather crudely, by the source-filter model [13]. A simplified way to visualise the vocal tract as a continuously varying cross-sectional area chamber. A useful model of this can be made using coupled tubes of different cross-section areas [2]. This concept

is shown in Figure 2.2 which shows of a sequence of tubes, each of which represents a different area of the vocal tract. Each tube has a different cross-sectional area denoted by A_k. The tubes have varying length across the vocal tract, denoted by L_k. Each tube in the model will have a set of resonant frequencies that depend on the length. Longer lengths have lower resonant frequencies, compared to shorter tubes with higher resonant frequencies.

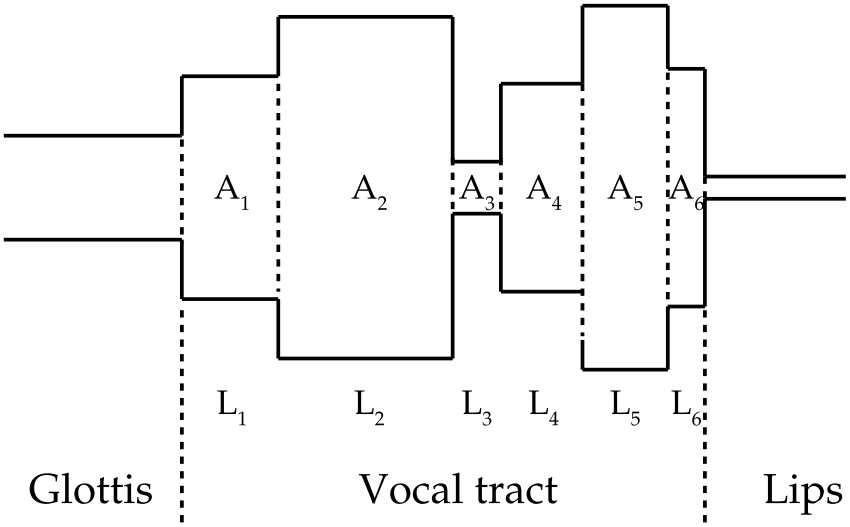


Figure 2.2: Acoustic tube model of speech production [2].

This model can be as complex or as simple as required. The more tubes in the model, the higher the resolution, and therefore the closer we get to the actual cross-sectional area vocal chamber, at the cost of a more complex model to work with. In continuous speech, the speaker can move the various articulators such as the tongue, lips and jaw in different configurations. At any point in time, this positioning can be approximated by the tube model. Just as the sound from a loudspeaker is modified changes according to the room/chamber it is transmitted in, the spectral properties of the sound waves change as they go through the vocal tract.

2.2 Front-End Preprocessing

Having discussed the physiological process of voice production, it is important to relate this to the mathematical extraction of features that map to the physical process of voice production. The primary operations for voice signal processing prior to feature extraction are digitization, pre-emphasis, frame blocking and windowing.

2.2.1 Digitization

Firstly an analogue voice signal has to be digitized to enable any kind of computational processing. During this step, the microphone and recording channels introduce undesired effects depending on their quality. For digitization theory and effects the reader is referred to [14].

2.2.2 Pre-emphasis

Some feature extraction systems pre-emphasise the signal before further processing. When acoustic energy is radiated through the lips it is subject to a boost of 6 dB/octave because of the radiation properties of the lips (compared to the glottal source for voiced sounds at a 6dB/octave slope). To counter this effect the signal is neutralised by a simple first-order filter that emphasizes the higher frequencies [15, 16] by an additional 6 dB/octave to reduce the effect of spectral tilt. The pre-emphasis filter in the time domain is shown in Equation 2.1 [17].

$$\tilde{s}(n) = s(n) - \lambda s(n-1) \tag{2.1}$$

The value of the pre-emphasis coefficient λ is in the interval [0.90, 0.98]. For fixed-point implementations a value of $\lambda = 15/16 = 0.9375$ is commonly used [17, 18].

2.2.3 Short-Term Frame Blocking

The properties of speech signals are statistically stationary over periods of about 10-30ms. Because of this, many approaches in speech signal processing are based on short-term analysis [19]. For this reason, a speech signal is blocked into frames of N samples of short duration in the 10-30ms range. The signal is then analysed one frame at a time, with frames advancing according to a "sliding window" with 30-50% overlap. An example of this process is shown in Figure 2.3.

The overlap is required in order to preserve any characteristics that are found in between frame boundaries. If a characteristic is present at a boundary and continues in another frame, then the overlap will allow gathering of this characteristic, as opposed to losing this information when no overlap is present. The amount of overlap therefore controls how quickly changes in parameters are noticed from frame to frame [20, 17].

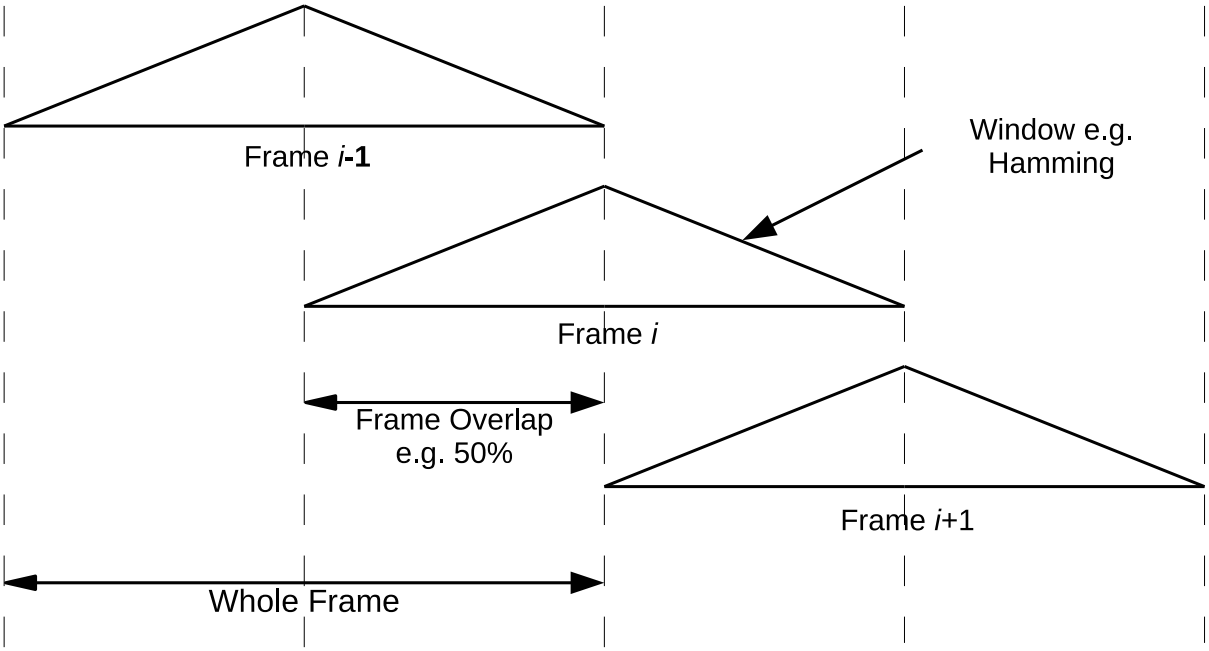


Figure 2.3: The short-term frame blocking process.

2.2.4 Windowing

The framing process described above introduces spectral artefacts, distorting some of the original spectral information in the Fourier transform. To reduce or avoid (when possible) this effect, windowing techniques are employed. In the time domain, a signal is point-wise multiplied by a window-weighting function. By convolution theory, this corresponds to convolution of the short-term spectrum of the frame with the window function magnitude spectrum response [21, 19, 22].

There are different window functions, and the choice of which is used is important for separating spectral components which are near one another in frequency or where one component is much smaller than another. For more details on window theory, the reader is referred to [20, 23]. Suffice to say that a good window function has a narrow main lobe and low sidelobe levels in their transfer functions. Every window function has a tradeoff between these two properties — a narrower main lobe increases side-lobe levels, and vice versa [21, 24]. When the Fast Fourier Transform (FFT) algorithm is applied to data (a fast implementation of the Discrete Fourier Transform (DFT)), spectral information from the FFT results occur at the wrong frequencies — the spectral information leaks over into adjacent frequency bins. This leakage is impossible to eliminate completely. However, the application of a window function reduces the most negative effects of spectral distortion [25]. The simplest windowing possible is no windowing at all,

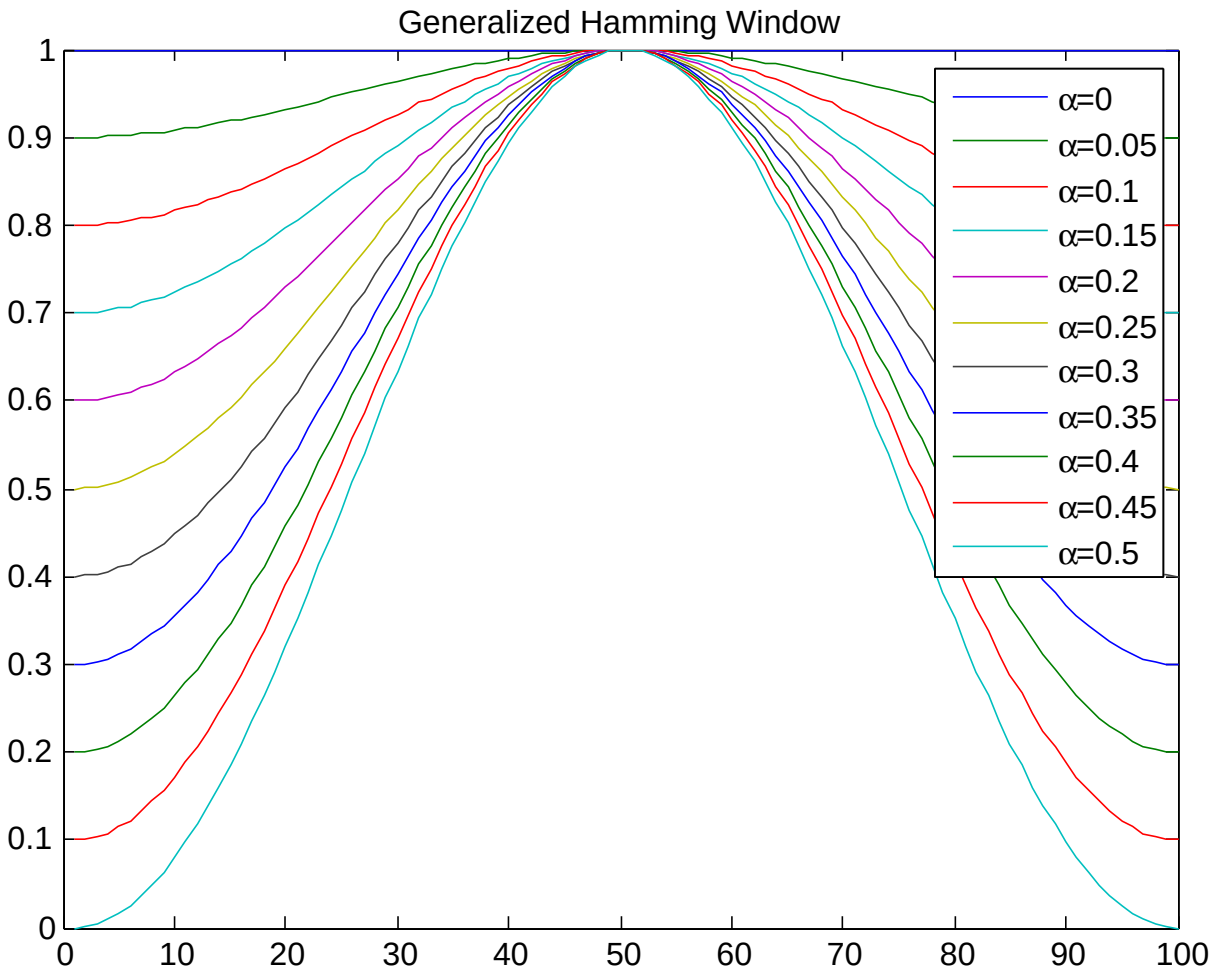


Figure 2.4: The generalized Hamming window with various values for α.

referred to as rectangular windowing, defined in Equation 2.2 [26].

$$w(n) = \begin{cases} 1, & 0 \leq n \leq N-1 \\ 0, & \text{otherwise} \end{cases} \tag{2.2}$$

A rectangular window keeps the original waveform unchanged and is not usually used, because it simply does not help with spectral leakage. Better windows are Hamming, Hann, Blackman, Bartlett and Kaiser windows. The generalized Hamming window is defined in Equation 2.3 [26].

$$w(n) = \begin{cases} \frac{\alpha-(1-\alpha)\cos(2\pi n/N)}{\beta}, & 0 \leq n \leq N-1 \\ 0, & \text{otherwise} \end{cases} \tag{2.3}$$

The value α is the window constant in the range $[0, 1]$, and N is the window duration in samples. To implement a Hamming window (our window of choice), the window constant is set to $\alpha = 0.54$, whereas to implement a Hann window, $\alpha = 0.50$. The value β is defined as the

normalization constant. Many implementations do not include this value as part of the equation. However, normalisation is important, so that the windowed function will give the same overall power of the signal as it was before windowing was applied (by modifying β such that the root mean square value of the window is unity). The value of β is defined in Equation 2.4 [17].

$$\beta = \sqrt{\frac{1}{N}\sum_{n=1}^{N} w^2(n)} \tag{2.4}$$

The generalised Hamming window can be seen in Figure 2.4, together with various values for α, including the special case for the Hann window.

2.3 Feature Extraction

After pre-processing, the voice signal is passed on to feature extraction modules. In this phase, each of the speech frames is converted to a low-dimension parameter-set that represents the acoustic information as a numerical vector. The algorithms used to obtain these feature vectors can be applied to both the time and the frequency domain. We describe the important techniques over the next section. When dealing with spectral domain (frequency) signals, the feature extraction methods available are more elaborate, and form the bulk of the signal processing that needs to be done in automatic speech systems. The spectral domain methods that we shall discuss all assume that the time domain signal has been transferred to the spectral domain via the DFT algorithm. It is beyond the scope of this chapter to discuss this algorithm here, but the interested reader is referred to [27].

2.3.1 Frame Energy and Power

Voice signals in the time domain can be thought of in terms of a function with varying amplitude through time. In this regard, the strength, or energy, of a signal can be measured by calculating the area of the function. However, voice signals have negative as well as positive amplitude values. The sections with negative amplitude do not have any less signal strength than sections with positive amplitude. In general, the power of a signal is proportional to its squared amplitude. Energy is the sum of the squared magnitude of all the digitised samples from 1 to N, as shown in Equation 2.5 [28].

$$E = \sum_{n=1}^{N} s^2(n) \tag{2.5}$$

It is also common to find the use of power, rather than energy as a feature, particularly the logarithm of the power, multiplied by 10, which is defined as the power of a signal in decibels. This is based on what is believed to be the logarithmic response of the human ear to audio. The signal being received by our brains, through our ears is not the same signal emitted from the source, and this has to be taken into consideration [28, 17].

2.3.2 Filter Banks

A signal has energy in many subbands (ranges in frequency), and we might wish to process each of these subbands independently, with different filters, instead of processing the entire signal under one filter. Feature extraction is then performed on the output of each subband filter [29]. This idea is simplified in Figure 2.5 [30].

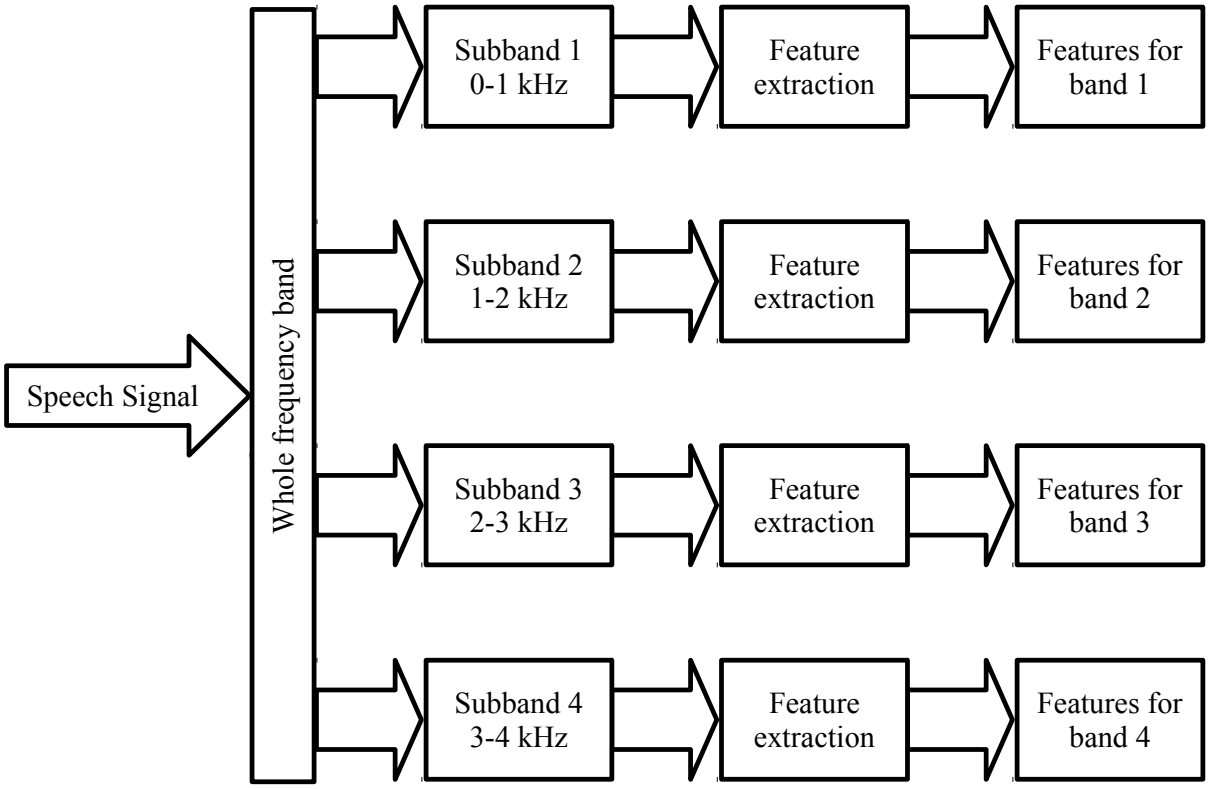


Figure 2.5: Subband-based feature extraction.

This technique was one of the first techniques ever specifically designed for speech signal processing, because it could be implemented easily with analogue circuits. It can also be implemented in the time domain using a set of recursive equations. However, its primary use is in the spectral domain. The advantage here is that once each subband is created form the signal, the same feature extraction techniques can be used as for a fullband signal, using regular frame-based processing [29]. Given that each subband is processed individually, the resolution

of the subband can be controlled more easily than in fullband processing [30]. In the frequency domain implementation, the processing is simply a multiplication of the signal spectrum with the filter magnitude response. Therefore, if we consider:

- an N-point magnitude spectrum (the spectral representation of a speech frame) $S(j)$, $j = 1, \ldots, N$
- an M-channel filterbank (M subbands) with sampled magnitude response specified in the arrays $H_i(j), i = 1, \ldots, M$

the output of the i^{th} filter $Y(i)$ is given in Equation 2.6 [31]. Every channel output is the frequency region (subband) weighted by the filter response (which can be any filter). The filter bank provides a tremendous drop in dimensionality ($M \ll N$).

$$Y(i) = \sum_{j=1}^{N} S(j)H_i(j) \tag{2.6}$$

An example of a filterbank magnitude response is shown in Figure 2.6. This filter bank is linearly spaced along the frequency range 0-8kHz. There are 20 filters (each colour coded differently), and every filter has a zero response outside of its passband. The triangular shape acts as a filter that changes the magnitude weight from 0 to 1 and back to 0 along the filter. In contrast, a rectangular filter bank would have a constant magnitude weight of 1 all along the filter. As a result, the output of the filtered subband would be the original information in the subband. A filterbank can be thought of as a simple model of the initial stages of the human auditory system, where frequencies within a certain bandwidth of certain frequency cannot be heard because of the phenomenon of "critical bands" [17, 32].

We have so far considered filter banks whose centre frequencies are linearly spaced. However studies have shown how frequencies perceived by humans are not linear with respect to the original source. Scales to model the psycho-acoustically motivated warping functions were proposed for the first time in 1937 by Stevens, Volkman and Newman, who proposed the mel scale. The mel scale is a scale of pitches as perceived by human listeners, and it shows that the perceived pitch of a human listener is not linear with respect to the real pitch emitted from a sound source [33]. This scale is shown in Figure 2.7 and can be computed using Equation 2.7.

$$f_m = 2595 \log_{10}\left(1 + \frac{f}{700}\right) \tag{2.7}$$

The mel scale gives an approximately linear response to frequencies below 1kHz. However, the

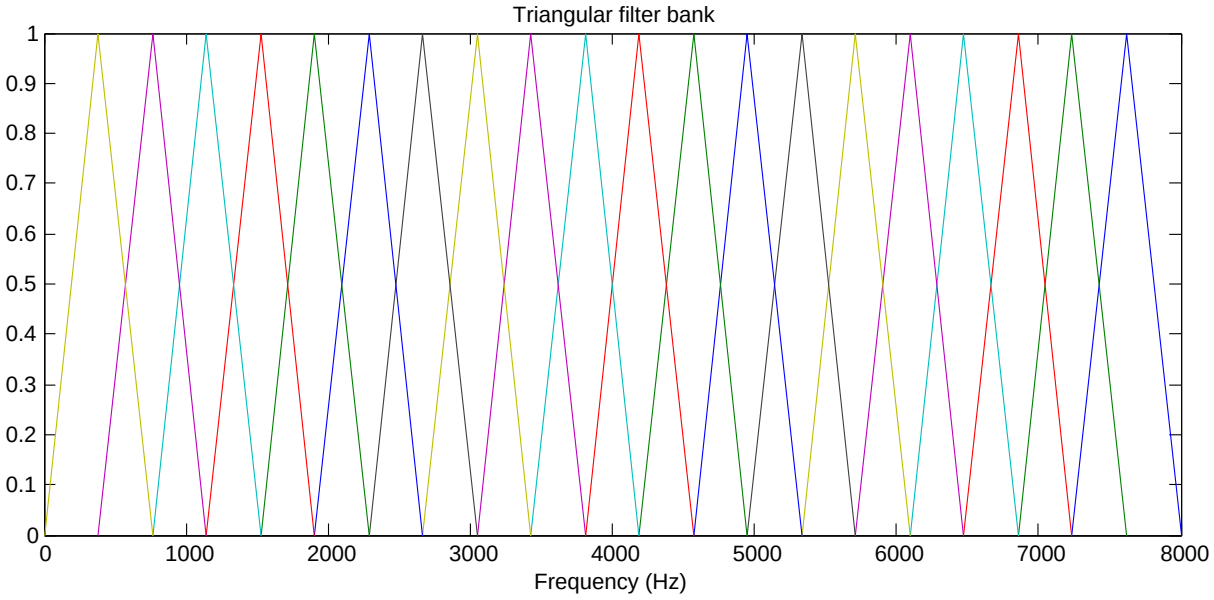


Figure 2.6: Linearly spaced triangular filter bank.

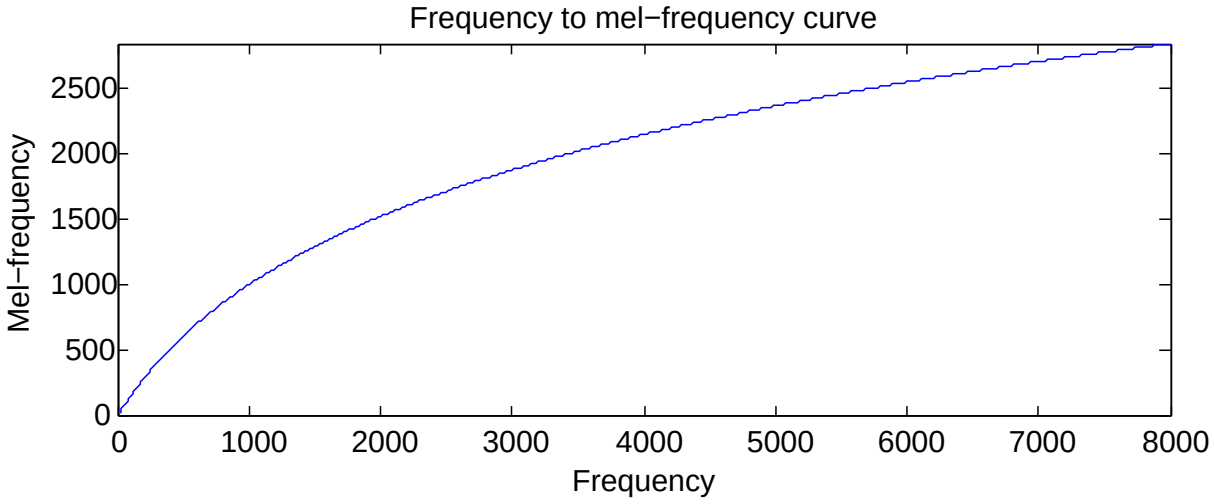


Figure 2.7: The mel scale curve.

response for frequencies above 1kHz is logarithmic. If we had to apply this frequency warping scale to the linear triangular filter bank used previously we would get the result shown in Figure 2.8.

Another important frequency warping scale used in speech technology applications is the Bark scale, proposed later on in 1961 by Barkhausen, which was a scale built on the perceived loudness of sounds by human listeners. The Bark scale is defined in Equation 2.8 [17].

$$f_b = 13 \arctan\left(\frac{0.76f}{1000}\right) + 3.5 \arctan\left(\frac{f^2}{7500^2}\right) \tag{2.8}$$

There are some reservations in the literature as to whether using these scales in speech

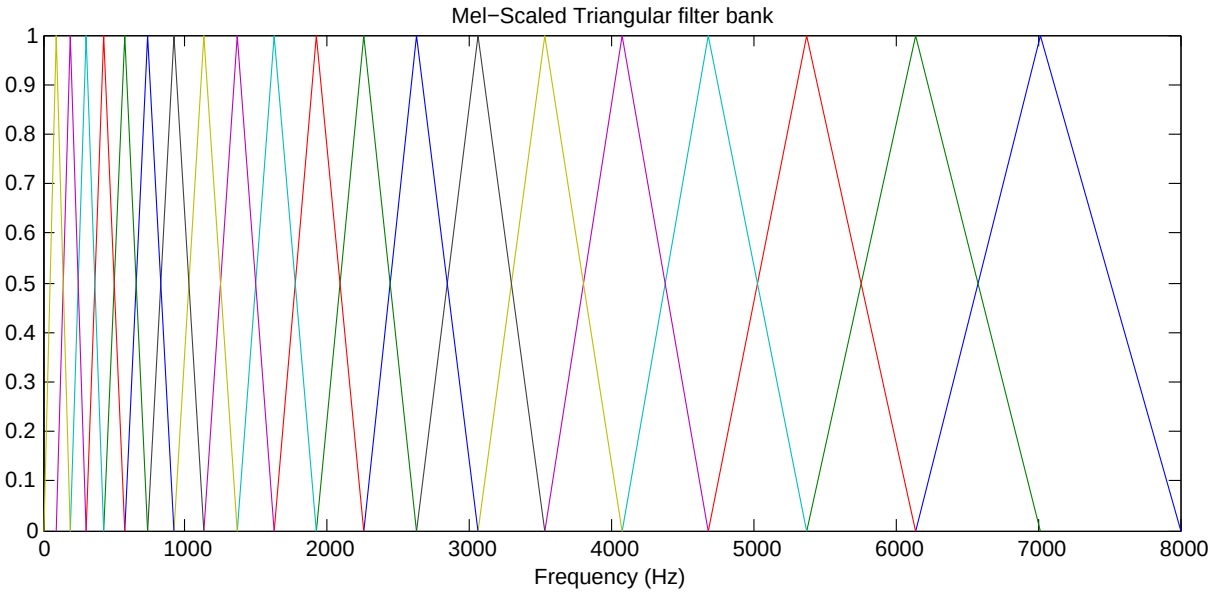


Figure 2.8: Mel scale warped triangular filterbank.

classification problems is advantageous. The argument is that there is no guarantee that the human auditory system is optimally designed for SID, AID etc., and for this reason, studies have been cautious about using these scales or not. By using these scales, the implicit assumption is that any information ignored by the human auditory system is not important for the speech systems. However, this may not be the case for some or all automatic speech systems, in that other data could be possibly useful computationally [34].

2.3.3 Cepstral Analysis

Having previously discussed how the vocal tract works via the acoustic tube model, we can discuss an important result that enables useful analysis via the spectral domain. The shape of the vocal tract can be estimated from the spectral shape of the emitted signal [2]. A voice signal can be represented as the convolution of a quickly varying source signal $e(n)$ (or excitation signal) with a slowly varying impulse response $h(n)$ of the vocal tract [22]. Therefore the voice production model we have described, would summarise the voice signal by time domain convolution as shown in Equation 2.9.

$$s(n) = e(n) * h(n) \tag{2.9}$$

The problem with this representation is that once the voice is recorded, we only have access to $s(n)$, the voice signal itself. It is desirable to separate the source signal (excitation) and the filter (impulse response of the vocal tract), so analysis can be performed on these components

individually. But with these components convolved non-linearly, it is very hard to extract this information. This problem is very much simplified in the frequency domain. The same result can be expressed in the spectral domain as shown in Equation 2.10.

$$S(f) = E(f)\,H(f) \tag{2.10}$$

By taking the logarithm of both sides, the result is the one shown in Equation 2.11, which dissolves the component into two additive parts [35].

$$\log|S(f)| = \log\left[|E(f)||H(f)|\right] = \log|E(f)| + \log|H(f)| \tag{2.11}$$

The spectral domain has transformed our signal to a linear multiplication (compared to convolution of the time domain) as show in Figure 2.9 [36], whilst the logarithm of the spectral domain converts the signal output to a summation of two distinguishable components. The log domain therefore provides us an additive superimposition, and the two components can be separated using conventional signal processing techniques.

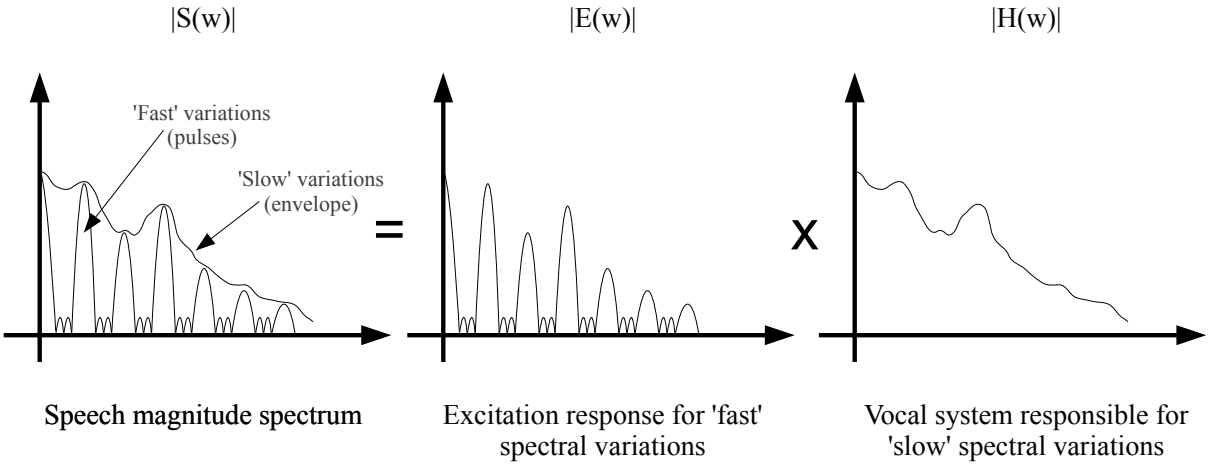


Figure 2.9: Spectrum components of a voice signal.

The techniques of cepstral analysis allows us to extract these two separate components. If the inverse Fourier transform is applied to the logarithm domain components, we have a mathematical guarantee that this will be applied individually to both of the components. This kind of processing is called *cepstral analysis*. Essentially, in cepstral analysis we are performing a frequency analysis of the spectral domain itself, creating a new domain called the *quefrequency* domain. The inverse Fourier transform will separate both the slowly varying and quickly varying parts of the signal, on different areas of what is called the quefrequency axis. This idea is demonstrated in Figure 2.10 [36].

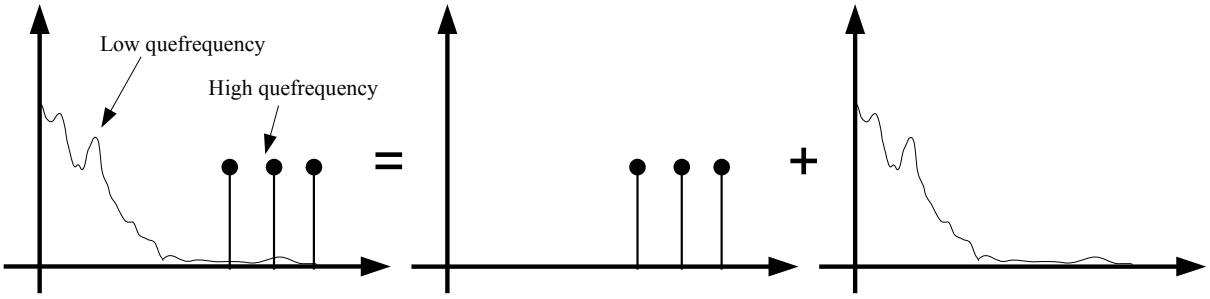


Figure 2.10: A voice cepstrum decomposition.

The low quefrequency terms of the cepstrum correspond to the slowly-varying properties of the voice signal, and hence represent the behaviour of the vocal tract for the particular voice frame. The high quefrequency terms on the other hand represent the quickly-varying characteristics of the voice frame, and describe the excitation pattern for the frame.

The resulting cepstrum is a vector of coefficients called cepstral coefficients. However, we have seen that the low quefrequency values are what are most important to describe the behaviour of the vocal tract. And for this reason, only the first few coefficients are important. The advantage in selecting only a subset of the cepstrum coefficients is that dimensionality is further reduced. The number of coefficients selected is usually in the order of 12 to 20 coefficients. The zeroth coefficient is usually dropped out because it represents the average log energy of the frame, and does not carry any speaker specific information [37, 38]. Having a maximum vector size of 20 coefficients to describe the vocal tract at a particular instant in time (the voice frame) is a very compact representation, and also very easy to work with.

We have previously described the psycho-acoustically motivated Mel and Bark scales. These scales can be used when performing cepstral analysis as well. The process is similar to cepstrum calculation, except that an extra step is inserted. The frequency axis is warped according to the particular scale prior to cepstral analysis [38]. The coefficients resulting from Mel scale warping are called MFCC. Some authors do not agree that the psychoacoustic analysis on which MFCC are based is suitable for problems such as SID [34]. However, in practice no feature set seems to beat MFCC in performance in SID [39, 40, 41].

2.3.4 Temporal Derivatives

Once absolute measurements such as MFCC have been extracted from the voice signal, it has become standard procedure to also add extra temporal information. The spectral parameters gathered via spectral feature extraction only gather characteristics of a particular instance in

time. However, when speaking, the articulators are continuously changing their positions, with specific rates of change. These rate of change of the articulatory movements depends on the speaking style, the speaking rate, and the speech context itself. At a lower level, these changes also depend on how the speaker blends various unit sounds together to form larger unit sounds such as diphthongs. It is usually desirable to capture these spectral dynamics.

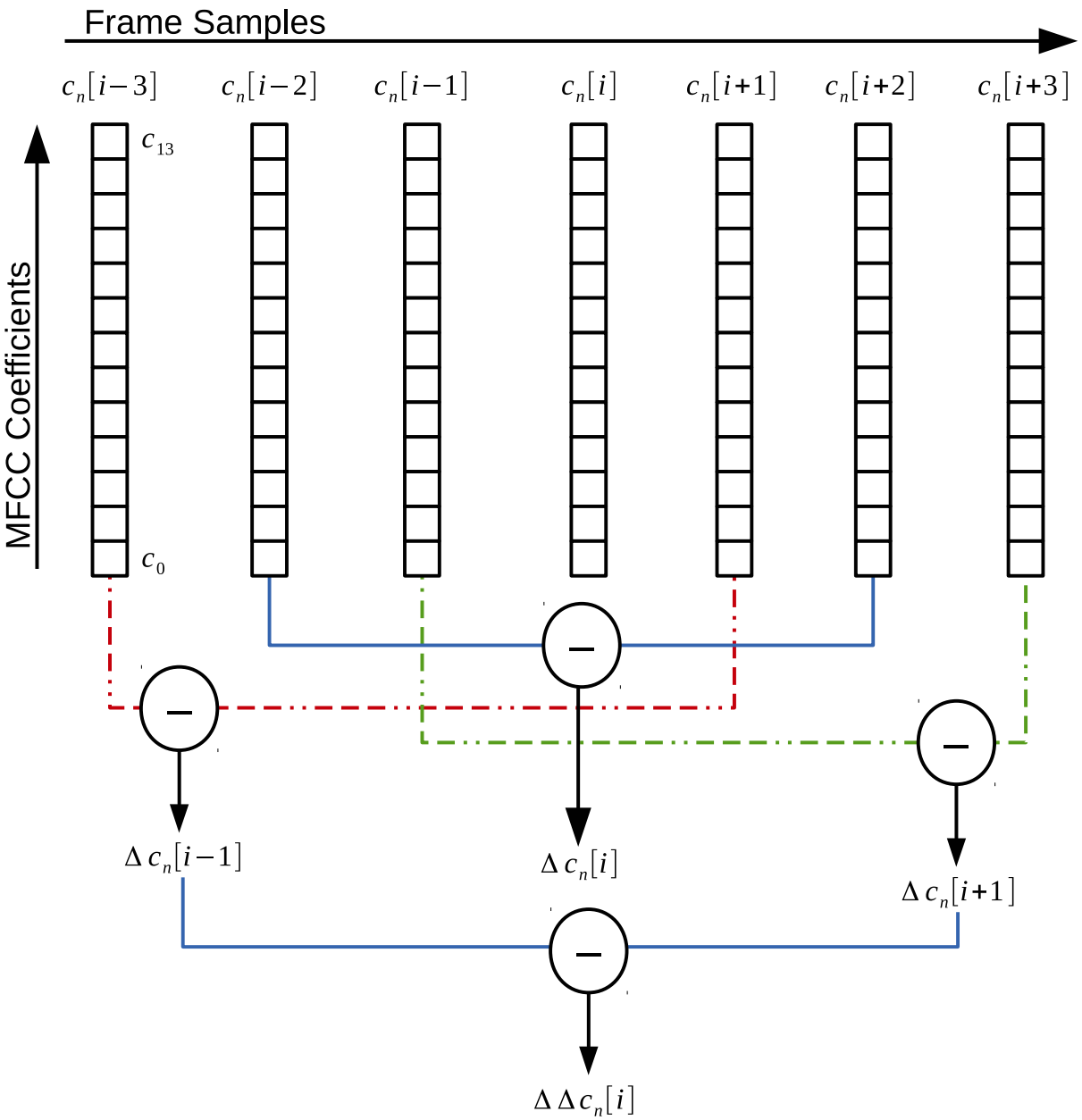


Figure 2.11: Deriving first and second order derivatives from absolute coefficients.

In order to do this, higher order time derivatives of the absolute measurements are captured. These time derivatives are known as delta-features [37]. If $c_n(i)$ denotes the i^{th} cepstral frame, the first order derivatives are defined in Equation 2.12. However, second-order derivatives can also be gathered similarly, by re-deriving from the first-order derivatives. This concept

is demonstrated in Figure 2.11. The first order derivatives and second order derivatives are most commonly referred to as delta and delta-delta parameters respectively. The value of d is usually assigned in the range of one to three frames. The process can be extended to higher order derivatives, however in speech signal analysis, first and second order derivatives are those used [17]. Note that this is the simple differences method of extracting temporal derivatives. The HTK book [42] provides the simple differences implementation, as well as an additional implementation, based on regression. However for the purpose of this thesis, we use the differences implementation for its simplicity.

$$\Delta c_n[i] = c_n[i+d] - c_n(i-d) \text{ where } n = 0, \ldots, N \text{ and } i = 0, \ldots, k-d \tag{2.12}$$

The output generated from spectral dynamics processing is a parameter vector that includes both the original absolute parameters as well as the temporal derivatives. Of course, the dimensionality of the vector space increases with every parameter that is added. Therefore constructing a reliable voice model for all the dimensions will require more training data. This fact should not be overlooked, because if training data is sparse, then it is probably better to leave out temporal derivatives altogether [37].

2.3.5 Shifted Delta Cepstra

The use of delta features can be found in many SID applications. For the problem of LID, a new feature set called Shifted Delta Cepstra (SDC) was introduced, and improved performance when compared to traditional delta features. [43, 44].

$$\Delta_{SDC} c_n[t,i] = c_n[t+iP+d] - c_n[t+iP-d] \text{ where } n = 0, \ldots, N \text{ and } i = 0, \ldots, k-1 \tag{2.13}$$

SDCs are an extension over temporal derivatives. They are constructed by combining the delta cepstra computed across multiple frames of speech. A SDC configuration is made up of four parameters with a notation *N-d-P-k*. N is the number of cepstral coefficients computed for each frame, d represents the look-ahead and look-back delay for the delta computation, k is the number of units for which delta coefficients are concatenated to form the final feature vector and P is the time shift between consecutive units. This concept is demonstrated in Figure 2.12. In this example the *N-d-P-k* are set as 7-1-3-7, which is a popular choice for LID [44, 45], resulting in a final SDC feature vector of 49 dimensions. If $c_n(t,i)$ denotes the i^{th} cepstral frame at time t, the SDC are defined in Equation 2.13.

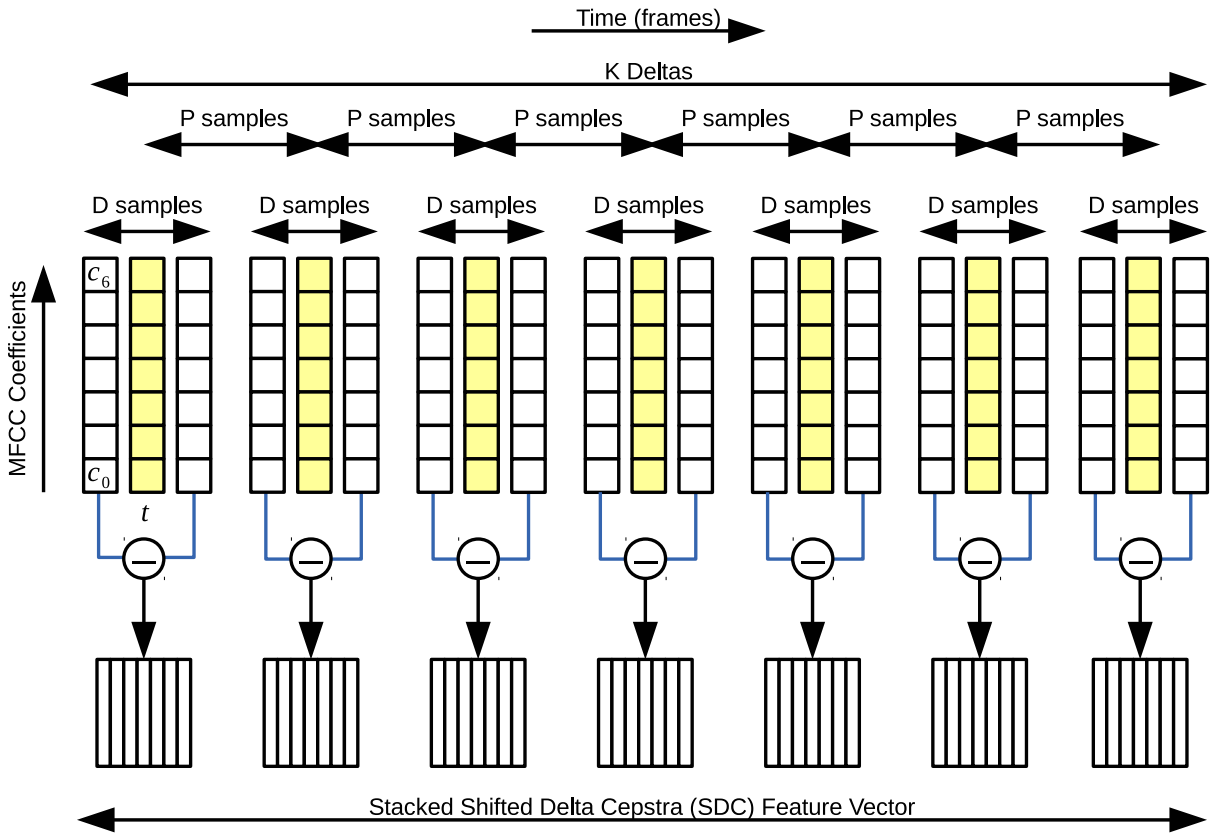


Figure 2.12: Deriving shifted delta cepstra from absolute coefficients in a 7-1-3-7 configuration.

2.3.6 Feature Normalization

The features collected from speech signals usually vary quite a bit due to noisy conditions and variations caused by channel differences. These effects can be very detrimental to classification systems such as GID, AID, LID, SID etc. There are a number of techniques that can be applied to the original features in order to attenuate these effects. The most common ones used are Cepstral Mean Normalization (CMN) [46], Mean and Variance Normalization (MVN) [47] and Feature Warping [48].

2.3.6.1 Cepstral Mean Normalization

In this feature normalization method there is an assumption that throughout the entire utterance, there is a stationary response present e.g. the frequency response of a specific microphone being used for recording. If this is so, the channel will effectively have been filtered by this transfer function throughout the entire utterance. We have seen earlier how convolution is equivalent to an additive component in the log domain, such as the log cepstral domain. By subtracting the mean cepstral vector from the whole utterance we remove the (stationary) offset dictated

by the channel. A variant of CMN adapts for varying channel effects, where CMN is applied to windows of speech within the utterance, rather than the whole utterance. Some slowly time-varying channel characteristics can be attenuated this way. An example of the effect of this technique on a cepstral component is shown in Figure 2.13. The overall shape of the distribution is unchanged, since the same mean vector is subtracted from all other vectors.

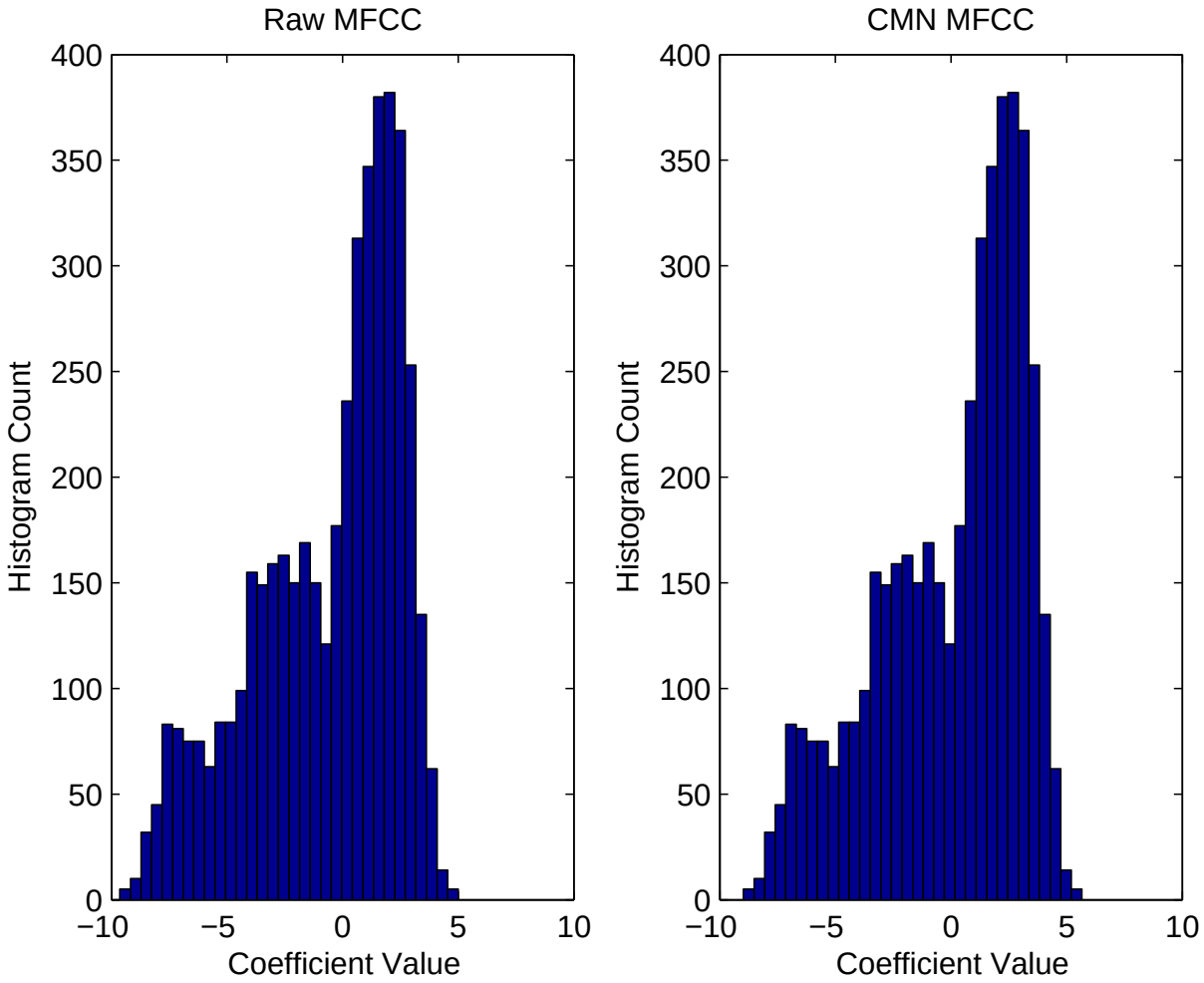


Figure 2.13: Cepstral mean normalization for c_1 over an utterance.

2.3.6.2 Mean and Variance Normalization

In this feature normalization method, the same process as in CMN is applied. In addition each cepstral vector is normalized by the variance cepstral vector of the whole utterance. The resulting features after MVN will have zero mean and unit variance, as opposed to just zero mean in CMN. An example of the effect of this technique on a cepstral component is shown in Figure 2.14. The shape of the distribution is also unchanged, however the entire variance is shifted (the x-axis variance is shifted to a common range for all vectors and all utterances). In addition to the

advantages offered by CMN, in this case, if features collected from different channel conditions etc. have different distributions, they are all remapped to the same distribution after variance normalization.

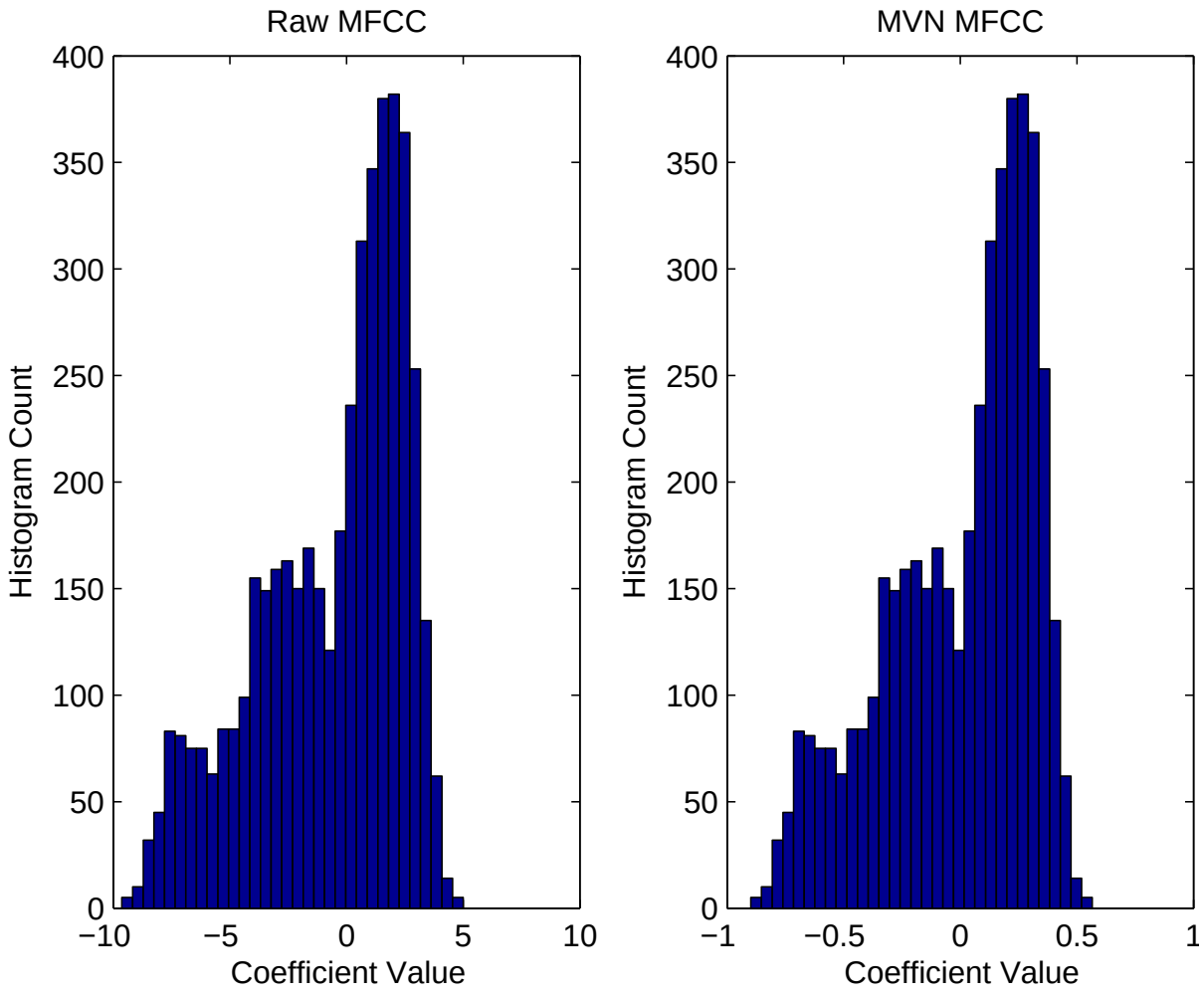


Figure 2.14: Cepstral mean and variance normalization for c_1 over an utterance.

2.3.6.3 Feature Warping

In this feature normalization method, individual cepstral features are conditioned to follow a specific target distribution over a certain time window duration. The short-term mean is implicitly also removed, and so the linear behaviour of the channel is removed. In addition to mean and variance, feature warping normalizes the flatness and skewness of the feature distribution. Additive noise effects are attenuated when the distribution shape is conformed to a particular distribution. Slowly changing additive noise can reduce the variance and distort or skew the distribution of spectral features. Feature warping is capable of conditioning this feature distribution by remapping the upper percentile of the source distribution to the upper

portion of the target (Gaussian) distribution, resulting in far more limited skew by noise. This method is similar to histogram equalization of picture pixel intensities. The target distribution of choice is generally the normal distribution with zero mean and unity variance. In addition the warping technique is performed over a window of three seconds in [48]. The choice of three seconds is loosely based on the assumption that over this short period, the underlying distribution of cepstral features is close to the normal distribution.

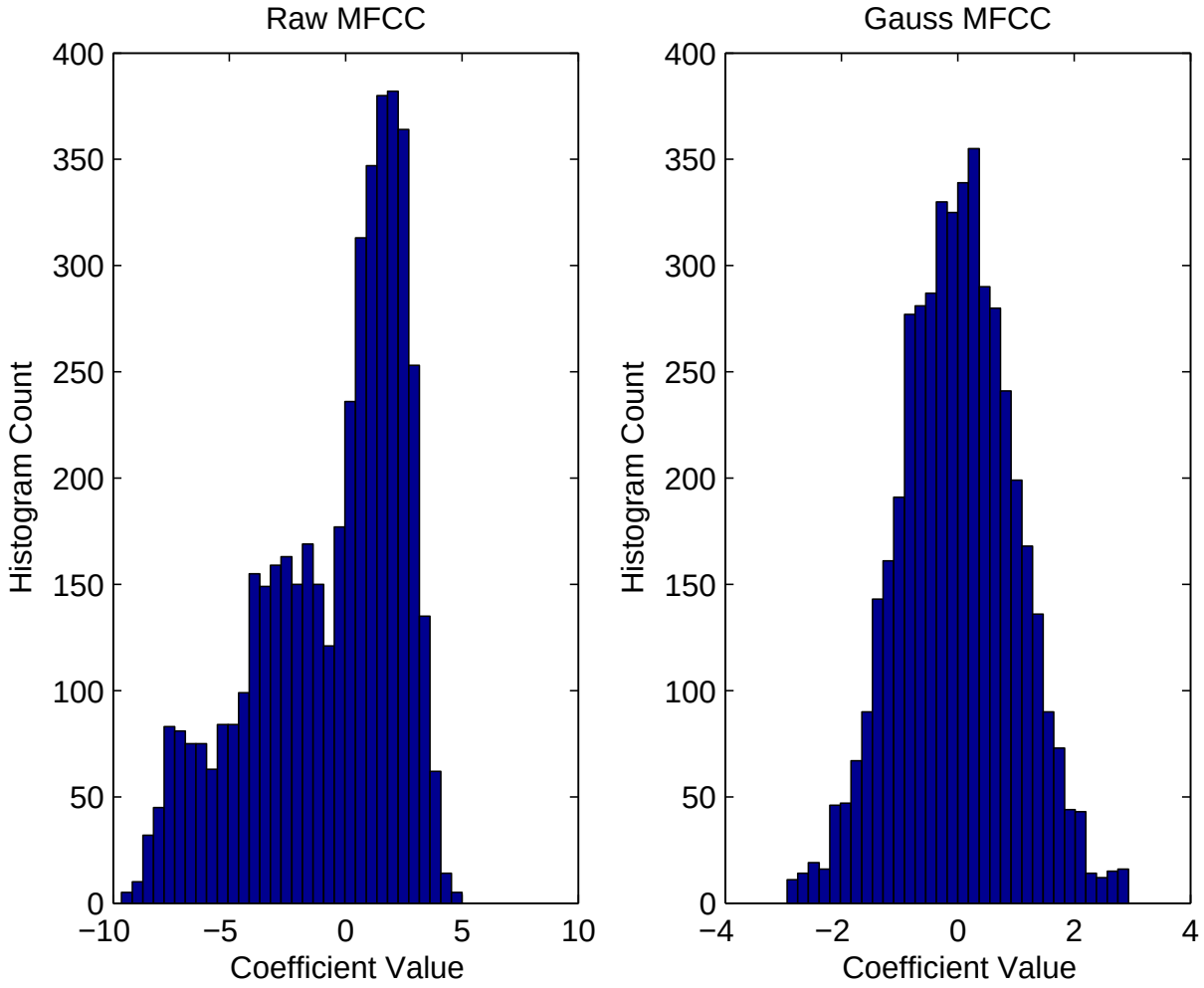


Figure 2.15: Feature warping (gaussianization) for c_1 over an utterance.

Given a window, only the central frame of the window is warped via a cumulative distribution function (CDF) to match the desired distribution. The features of a given window are sorted in descending order and ranked, with the most positive value obtaining a ranking of 1, while the most negative a ranking of N where N is the number of features in the window. This ranking is used as an index into a CDF lookup table for the corresponding warped feature value. The lookup table is calculated via Equation 2.14, where $h(z)$ is the target distribution. The warped feature can be determined by finding m in this equation. An example of the effect of this technique on a cepstral component is shown in Figure 2.15. Here the shape of the distribution

is entirely different (Gaussianized), and so is the variance range, which is centred around the mean for all utterance vectors to which this process is applied.

$$\frac{N + \frac{1}{2} - R}{N} = \int_{z=-\infty}^{m} h(z)dz \tag{2.14}$$

2.3.7 Time Domain Fundamental Frequency Estimation

A very important vocal feature is fundamental frequency (or pitch period). By approximating the pitch behaviour of a speaker, we can identify groups of speakers with different pitch characteristics, such as in GID. Grouping would allow for a logical segregation of speakers, and the models constructed say, for AID or SID would be specific to a particular gender.

The fundamental frequency (F_0) is the frequency at which the vocal cords vibrate during a voiced sound, and it is usually estimated on a logarithmic scale to match the resolution of how humans perceive pitches. The range of fundamental frequencies for both male and female voiced sounds lies in the range 50 Hz $< F_0 <$ 500 Hz. On the other hand, for unvoiced speech, where the vocal folds do not vibrate periodically, pitch is undefined, and implemented as $F_0 = 0$ [49].

If we assume a periodic signal, then the frequency of oscillation is the inverse of the period of oscillation. However, as more components are added to a simple waveform, the concept of a main signal frequency is no longer clear. There are methods that attempt to derive F_0 directly from the time domain. The reasoning is that if a waveform is periodic, then there are time-repeating events that can be extracted and counted to derive F_0. However, the main difficulty with time domain analysis is that complex waveforms such as voice data very rarely have one event per cycle that can be extracted. On the other hand, time domain methods are mostly simple to understand and implement, and they are very computationally efficient [49].

The most successful time domain methods are those based on the autocorrelation properties of voice signals, particularly the short-time average magnitude difference function (AMDF). This is defined in Equation 2.15 [17].

$$R_D(k) = \sum_{n=1}^{N-k} |s(n) - s(n+k)| \qquad \text{k=0,1,. . . ,N-1} \tag{2.15}$$

The value of k represents the lag (delay) time between the acoustic waveform and the copy of itself. As the time lag increases to equal the period duration of the short-time frame, the

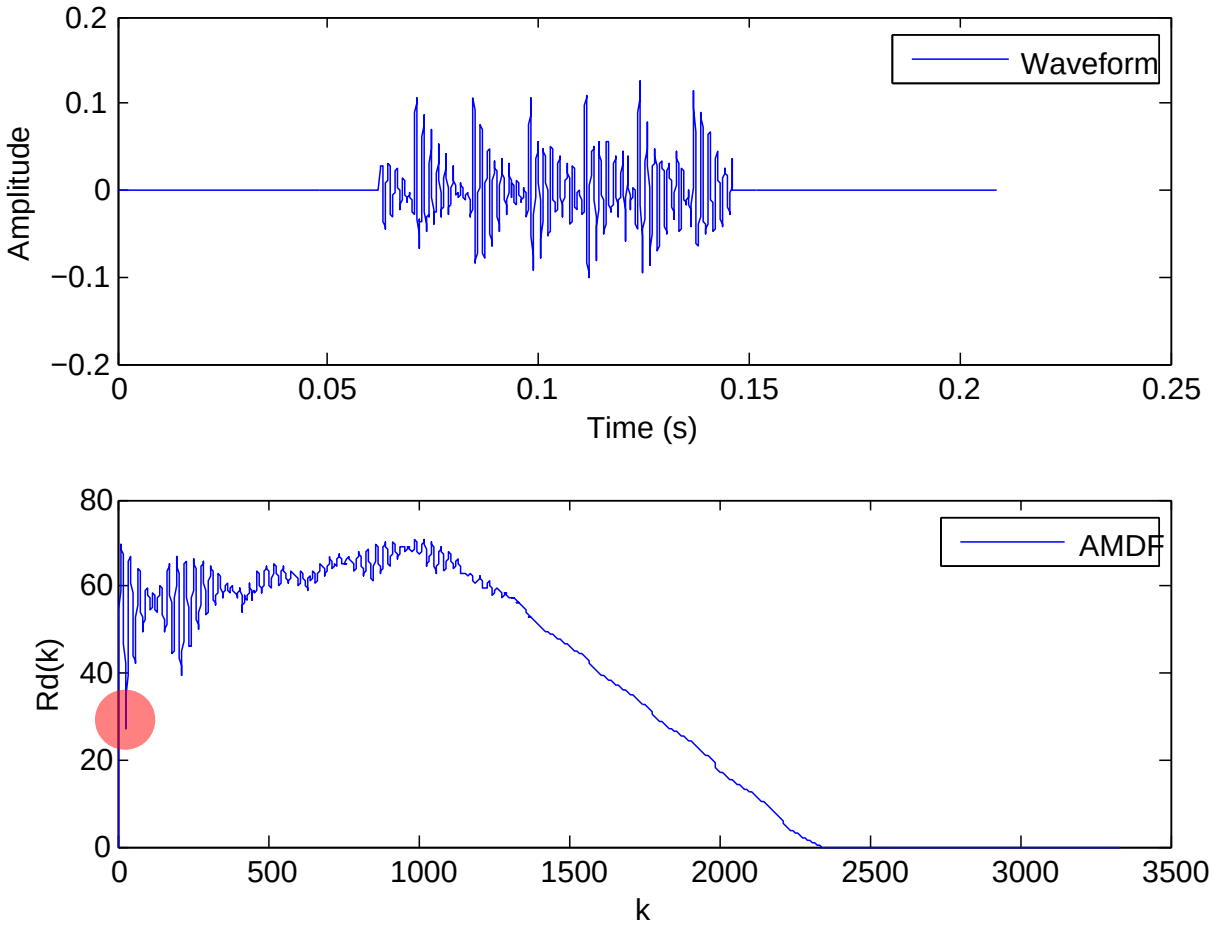


Figure 2.16: Fundamental frequency estimation via the AMDF algorithm.

correlation decreases to a strong minimum value (k_{min}) because the waveform is completely out of phase with its time-delayed copy [49]. The fundamental frequency can then be calculated as shown in Equation 2.16 [17].

$$F_0 = \frac{f_s}{k_{min}} \tag{2.16}$$

This is better shown through a demonstration. In Figure 2.16 we are shown a vowel 'a' in the time domain by a male speaker. The AMDF for this speech frame is in the second plot. The minimum value for k is found at the first local maximum, which in this case is $k = 30$. Given that f_s for this sample is 8000 Hz, then $F_0 = 266$ Hz.

The AMDF algorithm is computationally fast. However AMDF is only accurate for highly periodic signals, and can result in false period detection with signals that are either noisy, or signals with less obvious periodicity [49].

2.3.8 Spectral Domain Fundamental Frequency Estimation

Having previously discussed fundamental frequency estimation in the time domain, we now turn to the same concept as presented in the frequency domain. Reliable methods of F_0 estimation in the frequency domain are based on cepstral analysis. We have already described how a cepstrum is the Fourier transform of the log of the magnitude spectrum of the input waveform. In a frequency spectrum of a voiced sound, naturally occurring periodic information is present but it is difficult to automatically extract the pitch period from this information. The cepstrum though, moderates this effect to a great degree, especially for speech signals which are spectrally rich and have evenly spaced characteristics in short-time segments [49].

If the spectral information contains regularly spaced peaks that are not linearly visible, the cepstrum will reduce the peaks (now called *rahmonics*) and scale their amplitude to a usable setting. The result is a periodic waveform over the quefrequency axis (which is very closely related to time). The period (distance between rahmonics) is related to the fundamental frequency of the signal [50] The peak of the cepstrum is found by reading the quefrequency value (in time) with the highest amplitude, and converting this value back to the spectrum equivalent, thus obtaining F_0.

This is better shown through a demonstration. In Figure 2.17 we are shown a vowel 'a' in the time domain by a male speaker. This frame of audio is then converted to the spectral domain (second plot), and cepstral analysis is performed to bring out the quefrequency domain. The peak of the cepstrum can be seen in the 0.012 to 0.014 range of the quefrequency axis. The peak value can be then mapped back to its equivalent value in the spectrum. In this case F_0 = 74.0741 Hz.

Having discussed the general idea of both time and spectral domain fundamental frequency estimation, this thesis makes use of the algorithm in [51] to perform estimation. Whilst it is beyond the scope of this chapter to describe this algorithm in full, it is worth noting that this algorithm attempts to overcome some of the general shortcomings of the techniques described here, by utilising a normalized cross-correlation function (NCCF) [52]. In this implementation, two versions of the time-domain speech signal are provided, one at the original sample rate and another at a significantly reduced rate. The NCCF is computed for the low sample rate signal and locations of pitch maxima are noted. A second-pass NCCF is operated upon the regions of interest on the high sample rate equivalent portions to improve location and amplitude estimates. Therefore this thesis ultimately bases pitch tracking on the actual speech signal rather

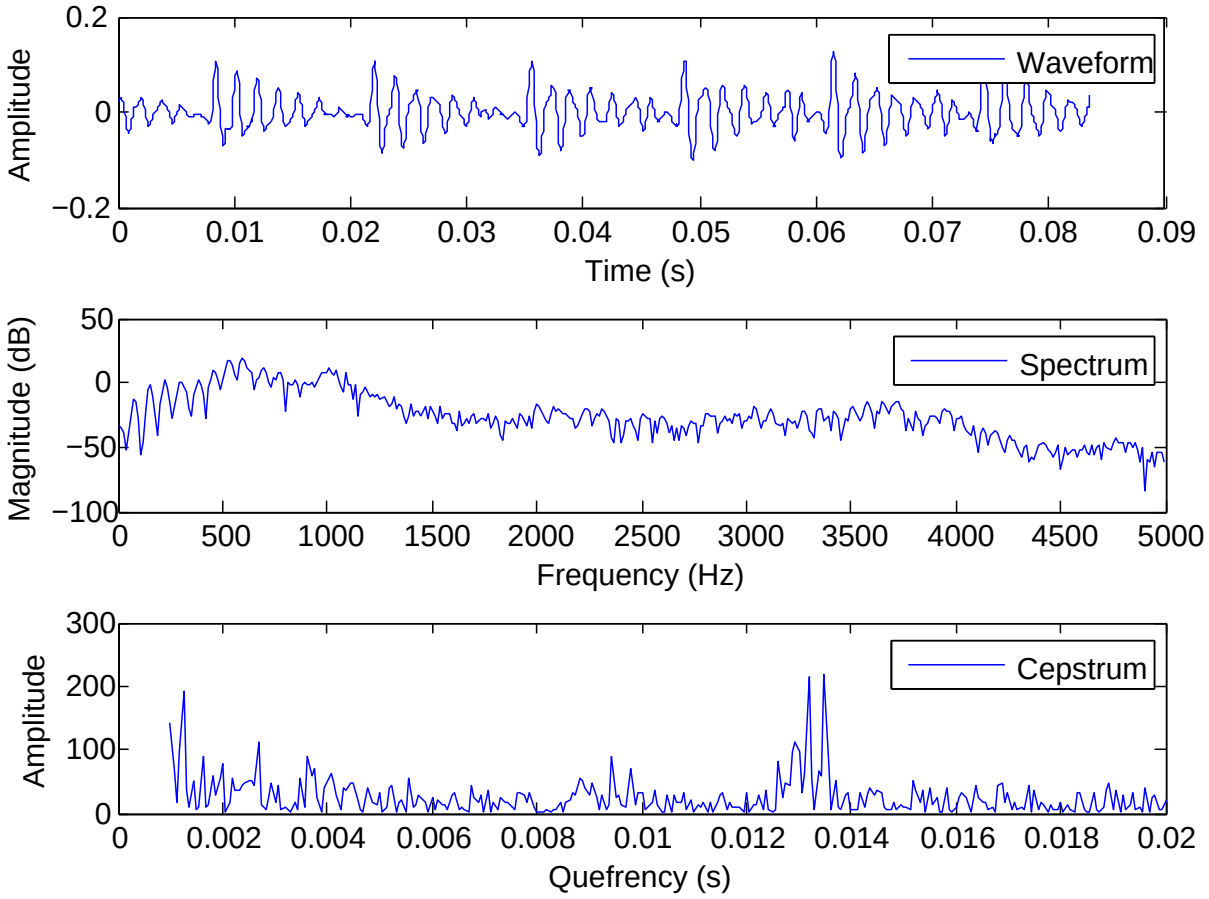


Figure 2.17: Fundamental frequency estimation via cepstral analysis.

than through cepstral analysis.

2.4 Feature Modelling and Classification

When feature vectors are generated for speech signals, the generic process that is required to build a speech system is that of modelling and classification. By modelling, we mean a way to characterise or distinguish between patterns amongst the various classes of data. In terms of a collection of speakers, classes can for instance refer to speaker groups of gender, accent, or at the lowest level, speakers themselves. A generic overview of this concept is summarized in Figure 2.18. The speech signals, and the respective feature vectors form part of the class from which they are collected. Classification methods usually depend on the modelling method being used, and are usually in the form of statistical inference based on the results obtained during the modelling stage. This chapter will cover core techniques underlying modelling and classification systems relevant to this thesis, with some reference to how they are conceptually applied to speech data. It is however, beyond the scope of this chapter to go through all the

possible variations available. The important variations related to this thesis will be dealt with in later chapters.

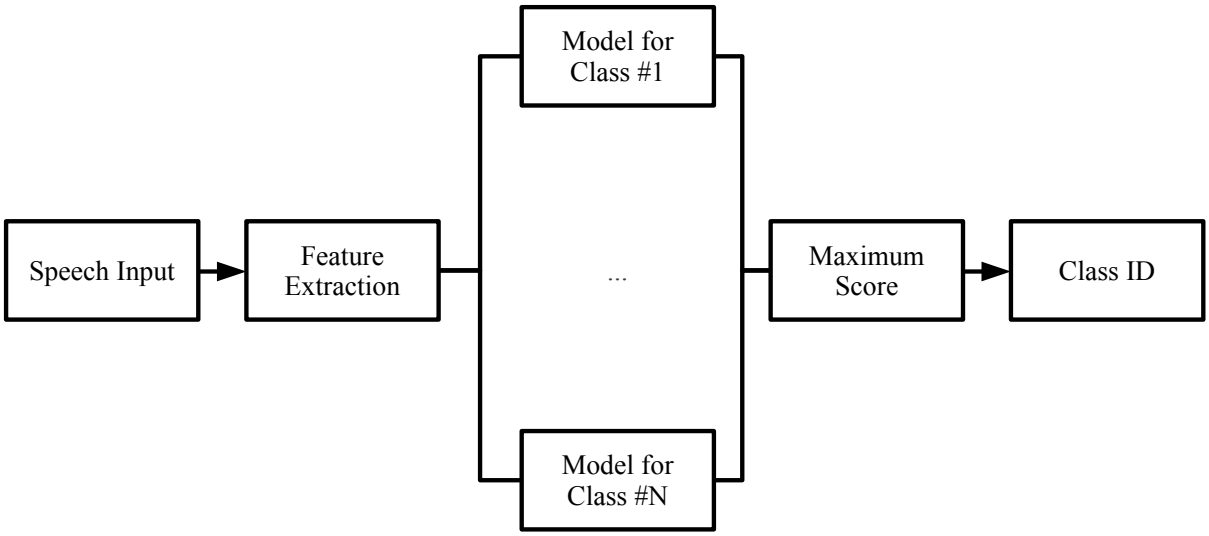


Figure 2.18: The feature classification process.

The task of classification for various class types can be split into two distinct approaches: text-dependent and text-independent methods. In text-dependent classification systems, training data is provided for for an utterance which is phonologically (though not necessarily phonetically) the same as that of the utterances used in the test phase. The techniques assume a direct dependency on training and test cases, and therefore each utterance is predefined. On the other hand, a text-independent system, does not have such dependencies and the training and test utterances can be totally different phonologically. Text-dependent systems directly exploit the features and cues associated with a predefined set of words and sounds. These systems are bound to achieve higher performing recognition rates. However, from both a computational and a forensic point of view, the applications of this technique are very much limited. In real life scenarios, we would not know a priori what the utterance should be. The aim is to go as far as possible in classifying voices and vocal traits from the acoustic cues, and to avoid explicit dependence on the content. This thesis deals with acoustic-only, text-independent techniques. However, our experimental results will later on be compared with text-dependent methods for completeness.

In text-independent classification, the utterance given by a speaker is not predefined. The words or sentences that are being spoken in the test phase are always unknown. A reference model for the class characteristics like gender, accent, speaker etc. is built on training samples, and because of the fact that the utterance is not predefined, the amount of data required for training usually has a strong bearing on the identification rate. For this reason, the acoustic

structure of a particular codeword cannot be exploited. Instead, reliable acoustic models covering as much as possible of the acoustic information (acoustic space) that a class can generate must be built with the available training utterances.

2.4.1 Vector Quantization

Vector quantization (VQ) models, or centroid models, are perhaps the simplest, and some of the most computationally efficient ways to model voice traits for text-independent classification. VQ reduces the data to a sequence of K symbols down from the original symbols comprising of the entire data set. Let us consider an acoustic class for which a number of vectors K representing short-term reference (training) utterances are available $R = \{\mathbf{r}_1, \mathbf{r}_2, \ldots, \mathbf{r}_K\}$. Each of these vectors has N dimensions, depending on the number of parameters in each vector. A traditional MFCC vector would have 13 dimensions, for example. The set of vectors R would take up a specific section of the vector space that is covered by the acoustical features for a particular class trait. Therefore every class model would ideally cover their own fraction of the vector space. When test vectors for an unknown class are acquired in the form $T = \{\mathbf{t}_1, \mathbf{t}_2, \ldots, \mathbf{t}_X\}$, they can be matched to the current vectors in the vector space, and the closest vector in N dimensions is identified. Therefore the class similarity and identification would be a result of finding the average distortion of the test vectors to each reference point, and selecting the reference with a lowest distortion value. Other techniques are possible, and this simple measure is called single nearest neighbour classification.

However, direct comparison of the test vectors to all the reference vectors for every class is computationally intensive, and mostly prohibitive for a large number of classes [53]. VQ therefore introduces codebooks to cluster the reference vectors into groups that are identified by a single centroid. The number of centroids would be considerably smaller than the number of reference vectors, speeding up the identification process by drastically reducing the number of comparisons required [54]. An example of a codebook for two clusters is shown in Figure 2.19.

Two choices must be made when using VQ for model training. The first is the decision on what clustering algorithm to use (since this has an effect on the resulting centroids), and the second is the size of the codebook (since every clustering algorithm requires this as an input parameter). A detailed study has been done in [55]. Six different clustering algorithms were analyzed for speaker identification (not accent):

1. Random: random codebook

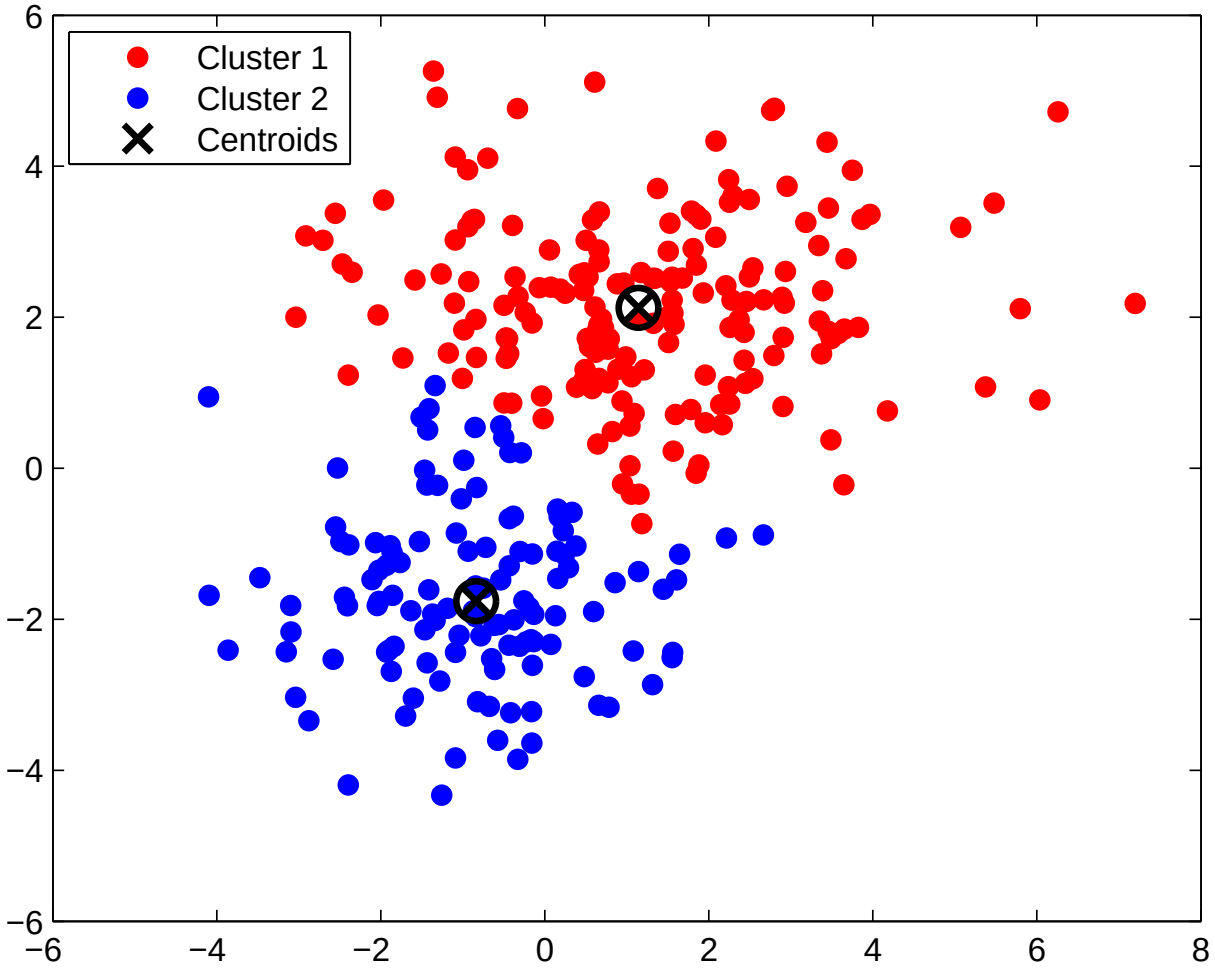


Figure 2.19: A dataset is reduced to a codebook of $K = 2$. The points associated with the different cluster centroids are colour-coded.

2. GLA: Generalized Lloyd algorithm [56]

3. SOM: Self-organizing maps [57]

4. PNN: Pairwise nearest neighbour [58]

5. SPLIT: Iterative splitting technique [59]

6. RLS: Randomized local search [60]

The work in [55] shows that the choice of clustering method has little, if any, effect on the correctness of identification. The centroids produced by these algorithms are only marginally different, making the corresponding recognition rates similar. The SPLIT algorithm is recommended for a large class database where running time is important, whilst the RLS algorithm is recommended for its implementation simplicity, and the slightly better results it achieves.

The most important task, though, is selecting an appropriate codebook size, which greatly

affects recognition rates. The best way to increase, and guarantee a good classification accuracy is to increase the codebook size high enough. Side-effects in increasing the codebook size are running-time, but also over-fitting, in cases where there are far too few samples for the codebook size. For a task such as speaker identification, the minimum recommended codebook size is 64. For 64 codes, four out of the six algorithms managed to achieve a 100% correct identification rate over the test speakers [55]. Increasing the codebook size to 256 did have some effect in that some algorithms performed better, whilst others performed worse. However, the average identification rate over all the algorithms remained the same.

Having a codebook for every class, and test vectors that require an identification, a measure of how distant the test vectors are from all the codebook vectors of a class is given by the average quantization distortion, shown in Equation 2.17, where $d(\cdot, \cdot)$ is a distance measure between two vectors. The choice of the distance measure is arbitrary. A common measure used is the Euclidean distance. Also, $D_Q(T, R) \neq D_Q(R, T)$ in practice due to a different number of test and reference vectors, making the relation a non-symmetric one [61].

$$D_Q(T, R) = \frac{1}{X} \sum_{x=1}^{X} \min_{1<k\leq K} d(\mathbf{t}_x, \mathbf{r}_k) \tag{2.17}$$

The smaller the average quantization distortion between T and R the more indicative it is that T and R originate from the same class. The centroid group for a class giving the smallest average quantization distortion identifies the class.

2.4.2 Mixture Models

The Gaussian Mixture Model (GMM) is a very popular and mature stochastic model. Stochastic models provide better flexibility and more meaningful results through probabilistic comparison [62]. GMM-based techniques have become the basis of many popular methods for modeling the distribution of vocal features in acoustic classification, and were originally introduced to speaker recognition by Reynolds [63, 64, 65].

We can represent the GMM by λ^n, as the model that represents the n^{th} class, which is built using the training data for the class. Therefore, for N classes, we will have N GMMs. As used previously the training utterance can be represented as a sequence of feature vectors as $R = \{\mathbf{r}_1, \mathbf{r}_2, \ldots, \mathbf{r}_K\}$, having K frames. In order to identify a class (from the group of classes N) from which the test utterance is generated, we need to identify the model that gives the maximum a posteriori probability for the observed utterance. So if the test utterance is represented

by $T = \{\mathbf{t}_1, \mathbf{t}_2, \ldots, \mathbf{t}_X\}$ having X frames, then the required model can be found by computing Equation 2.18 [62].

$$\mathcal{N}^* = \arg \max_{1 \leq n \leq N} P(\lambda^n | T) = \arg \max_{1 \leq n \leq N} \frac{P(T|\lambda^n)P(\lambda^n)}{P(T)} \tag{2.18}$$

The derivation follows from Bayes' rule of probability. The classes are all equally likely to have generated the test utterance (In our testing conditions we expect all test regions to be represented equally. Of course, in reality, one could factor in populations and conditions which would mean that this assumption would have to be compensated for in real operation.), and therefore $P(\lambda^n) = 1/S$. Also the value of $P(T)$ is an equal value for all the class models (this is an assumption we make, as we expect a balance dataset), and therefore we can summarize the relation as shown in Equation 2.19. Once a probability for each model is given, the class is identified by the highest scoring GMM, since the density of a GMM is an indication as to how close the model fits the observed (test) data. This reasoning is summarized in Equation 2.19 [62].

$$N^* = \arg \max_{1 \leq n \leq N} P(T|\lambda^n) \tag{2.19}$$

The task we need to perform is to compute a suitable model λ (we drop the superscript n for clarity) for each class. GMM models are conceptually very similar to VQ methods, and are often considered to be the stochastic extension to the centroid based clustering methods. The general idea is to have overlapping cluster boundaries. A feature vector would not be assigned to just one centroid, but rather, to all clusters, with a different non-zero probability for each cluster. Therefore every vector is a member of every cluster, but to a different degree of strength.

A GMM is built up by assuming that a set of feature vectors X is a linearly weighted finite mixture of M multivariate Gaussian probability density functions (PDFs). This is shown in Equation 2.20.

$$P(\mathbf{t}|\lambda) = \sum_{m=1}^{M} P_m \mathcal{N}(\mathbf{t}|\boldsymbol{\mu}_m, \Sigma_m) \tag{2.20}$$

where P_m is the prior probability (or mixture weight) of the m^{th} Gaussian, since each Gaussian in the mixture can be theoretically assigned a different weight. The term $N(x|\mu_m, \Sigma_m)$ is the Gaussian density function with D dimensions, and is defined in Equation 2.21 (adapted from [66]), with mean vector μ_m and covariance matrix Σ_m.

$$\mathcal{N}(\mathbf{t}|\mu_m, \Sigma_m) = \frac{1}{(2\pi)^{\frac{D}{2}} |\Sigma_m|^{\frac{1}{2}}} \exp\left[-\frac{1}{2}(\mathbf{t} - \boldsymbol{\mu}_m)^T (\Sigma_m)^{-1} (\mathbf{t} - \boldsymbol{\mu}_m)\right] \tag{2.21}$$

For simplicity, we denote a class GMM by $\lambda = \{P_m, \mu_m, \Sigma_m\}$ for $m = 1, \ldots, M$. What needs to

be done at this point is find proper values for the number of mixtures M, the mean vectors, the covariance matrix and the weights, such that the GMM properly fits the observed training data. An example of how the previous VQ codebook is represented parametrically by a GMM is shown in Figure 2.20.

Most spoken languages are made up of 30 to 40 phoneme units. Since the target of a GMM is to fit a model covering this space, a value larger than 32 is usually used for M. Also, for computational reasons, the covariance matrix is taken to be a diagonal one [67]. The reason is that using a full covariance matrix requires a lot of training data (that is usually unavailable) and is very expensive computationally. For examples of estimation using full covariance, the reader is referred to [68].

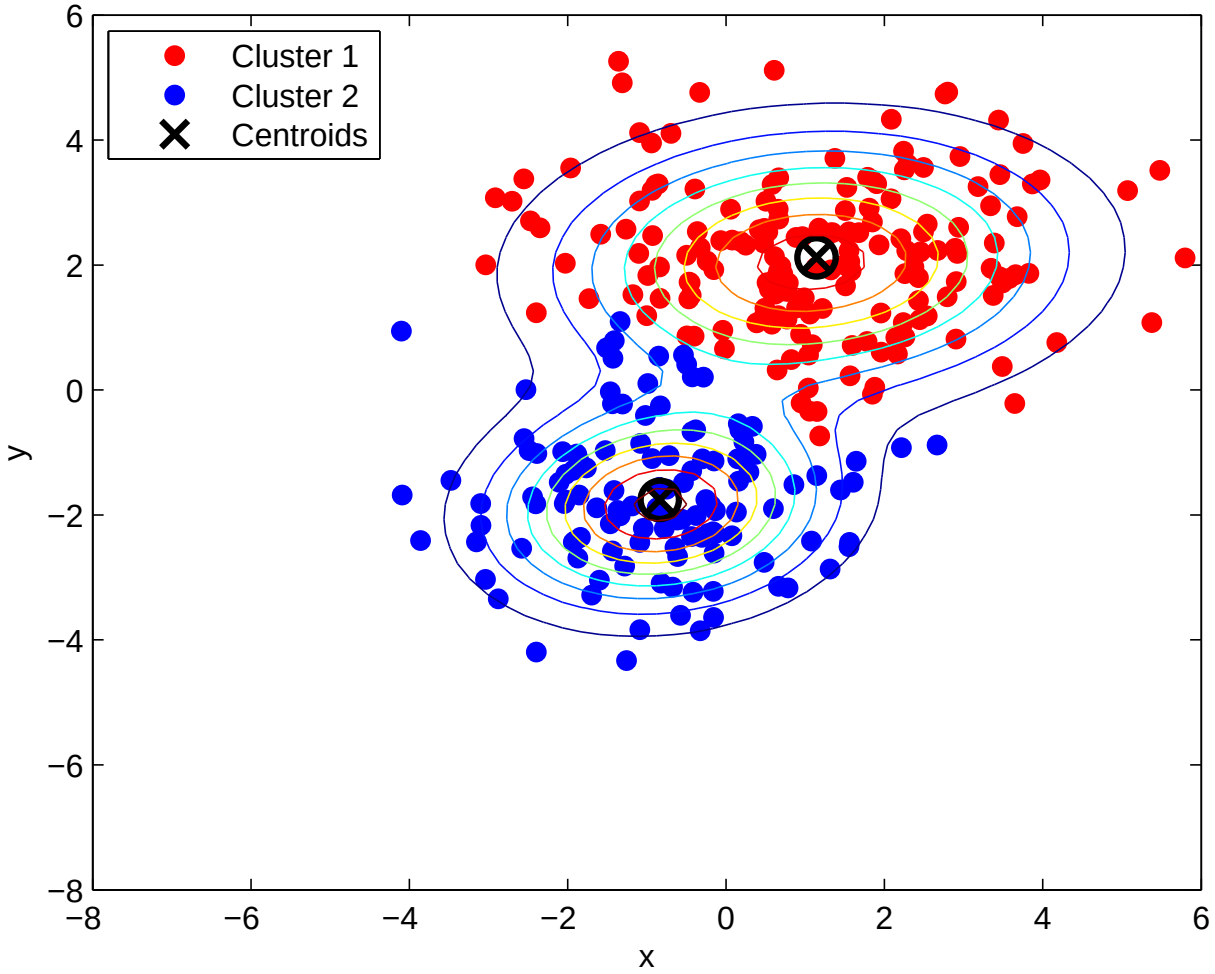


Figure 2.20: A dataset is reduced to a codebook of $K = 2$. The points associated with the different cluster centroids are colour-coded. The dataset is also defined in parametric form by density contours of a GMM which has been designed to represent the data by two Gaussian mixtures. The mean of the Gaussian mixtures are the codebook means. The covariance of each component in the mixture defines the general shape of the Gaussian.

2.4.2.1 Maximum Likelihood Training

In order to train a GMM, the maximum likelihood (ML) approach is used. The average log-likelihood for a set of reference vectors R for a GMM λ is given by Equation 2.22 [39].

$$LL_{avg}(R, \lambda) = \frac{1}{K} \sum_{i=1}^{K} \log \sum_{m=1}^{M} P_m \mathcal{N}(\mathbf{r}_k | \boldsymbol{\mu}_m, \Sigma_m) \tag{2.22}$$

A higher average log-likelihood value indicates that the model is more accurately modelling the data points in our reference vectors. To optimize and maximize this value, the Expectation-Maximization (EM) algorithm is used [67, 69, 70]. It consists of two steps: an E-step (Expectation) and a M-step (Maximization).

The GMM model parameters for this class are initialized using a clustering algorithm such as k-means, or other algorithms used in VQ. The EM algorithm for computing the GMM model parameters for a class is given below. Note that we have dropped the class specific superscript n for clarity.

1. The E-Step: Posterior probabilities are calculated for all the training feature vectors of the given class model λ using Equation 2.23.

$$P(m|r_n, \lambda) = \frac{P_m \mathcal{N}(\mathbf{r}_n | \boldsymbol{\mu}_m, \Sigma_m)}{\sum_{i=1}^{M} P_i \mathcal{N}(\mathbf{r}_n | \boldsymbol{\mu}_i, \Sigma_i)} \tag{2.23}$$

2. The M-Step: The M-Step uses the posterior probabilities from the E-Step to estimate model parameters by using Equations 2.24, 2.25 and 2.26.

$$\hat{P}_m = \frac{1}{K} \sum_{k=1}^{K} P(m|\mathbf{r}_k, \lambda) \tag{2.24}$$

$$\hat{\mu}_m = \frac{\sum_{k=1}^{K} P(m|\mathbf{r}_k, \lambda)\mathbf{r}_k}{\sum_{k=1}^{K} P(m|\mathbf{r}_k, \lambda)} \tag{2.25}$$

$$\hat{\Sigma}_m = \frac{\sum_{k=1}^{K} P(m|\mathbf{r}_k, \lambda)(\mathbf{r}_k - \hat{\boldsymbol{\mu}}_m)(\mathbf{r}_k - \hat{\boldsymbol{\mu}}_m)^T}{\sum_{k=1}^{K} P(m|\mathbf{r}_k, \lambda)} \tag{2.26}$$

3. Set $P_m = \hat{P}_m$, $\boldsymbol{\mu}_m = \hat{\boldsymbol{\mu}}_m$, and $\Sigma_m = \hat{\Sigma}_m$ and iterate the sequence of E-Step and M-Step a few times till convergence is reached. On each iteration of the EM algorithm, the variance is limited by a variance floor to reduce singularities in the final model [40]. Only a small number of iterations (or none at all) are required for the algorithm to converge [71, 72, 73].

One important consideration to make when building these acoustic models, is that the acoustic environment and recording channels can vary between the training and testing phase. Therefore, if we are to build a statistical model representing the acoustical behaviour of a class, we must make sure that the models can adapt well to new recording channels and environments. For this reason, a system to model the variability of different speakers and environments is built, in a similar fashion to models for individual class. This model will be built using hours of speech from many different classes, to represent cohort speakers and environment variability, and is termed as the Universal Background Model (UBM) [74].

2.4.2.2 Maximum a Posteriori Adaptation

Given that the UBM represents class-independent voice features, we can state that the UBM has prior knowledge about the feature distribution of the general class parameters. As a result, UBMs have been used for more efficient construction of class specific models. Instead of computing a new GMM from scratch for every class, the UBM is used a starting estimate of the class specific GMM. The parameters are then adapted to model the differences for a particular class. Practice has shown that this method is more efficient computationally, and that the accuracy of results is equivalent [74]. Some studies report that it is advantageous to build gender-dependent UBMs and extracting new class models based on gender, since male and female acoustic properties can vary widely on parametric models such as GMMs [75].

Adapting the UBM model to create a class-specific model is done via the maximum a posteriori (MAP) algorithm [74]. If we consider the training samples $X = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_T\}$, and the UBM defined as $\lambda_{UBM} = \{P_m, \mu_m, \Sigma_m\}$ for $m = 1, \ldots, M$, then we can define the new adapted mean vector $\hat{\mu}_m$ by Equation 2.27. The adaptation depends on the relevance parameter r, which controls the degree of effect of the training samples on the UBM mean vectors [74, 39].

$$
\begin{aligned}
\hat{\boldsymbol{\mu}}_m &= \alpha_m \hat{\mathbf{x}}_m + (1 - \alpha_m)\boldsymbol{\mu}_m \text{ where} \\
\alpha_m &= \frac{n_m}{n_m + r} \\
\hat{\mathbf{x}}_m &= \frac{1}{n_m} \sum_{t=1}^{T} P(m|\mathbf{x}_t)\mathbf{x}_t \\
n_m &= \sum_{m=1}^{T} P(m|\mathbf{x}_t) \\
P(m|\mathbf{x}_t) &= \frac{P_m \mathcal{N}(\mathbf{x}_t|\boldsymbol{\mu}_m, \Sigma_m)}{\sum_{i=1}^{M} P_i \mathcal{N}(\mathbf{x}_t|\boldsymbol{\mu}_i, \Sigma_i)}
\end{aligned}
\tag{2.27}
$$

Once the class specific models are created, class identification becomes a trivial task. The matching score depends on both the class-specific model and the UBM. In order to identify a class, the difference of average log likelihood ratios is calculated as shown in Equation 2.28.

$$LLD_{\text{diff}} = \log P(X|\lambda_s) - \log P(X|\lambda_{UBM}) \tag{2.28}$$

This difference is calculated for every class in the set, and the highest average log likelihood difference score would identify the class. Class verification is also done in the same manner, where instead of calculating a score over all the class set, only the claimed class model is used, and the resulting score is compared to a threshold value Θ. The claimed class is accepted if $LLD_{\text{diff}} > \Theta$ and rejected otherwise.

2.4.3 Support Vector Machines

The classification methods we have considered so far consider feature vectors that model the distribution of classes. These methods are termed *generative classifiers*. Another way of modelling classes is with the use of *discriminative classifiers*, where the model focuses on what discriminates one class from another. With this philosophy in mind, support vector machines (SVMs) [76, 77, 78] have proved to be a very good alternative to GMMs for acoustic classification problems, especially in classifying unseen data, or relevant test data that differs a lot from the matching reference examples [45, 79].

SVMs are primarily a binary classifier, where a sample is classified as either belonging to a class, or not. In training, SVMs need to be supplied with labels denoting positive or negative membership to a class. Formally, for a training set of size N, with dimensionality D, and labels $+1$ for positive examples and -1 for negative examples, then the training set is a tuple x_i, y_i, where $i = 1, 2, \ldots, N$ and $y_i \in \{-1, +1\}, x \in \mathbb{R}^D$. We shall go over some important properties of SVMs in detail to give some intuition about the properties of the discriminative mechanism for this model.

An SVM is a binary classifier that separates two sets of data. One set of data represents the positive reference vectors of the target class (green), while the other set of negative data (red) would represent all reference vectors not belonging to the target class. The task of an SVM is to find a hyperplane (decision surface) between the positive and negative class examples. This is demonstrated in Figure 2.21 for two dimensions (the separator is a line) and Figure 2.22 for three dimensions (the separator is a plane). In general, if the data is in N-dimensions, the

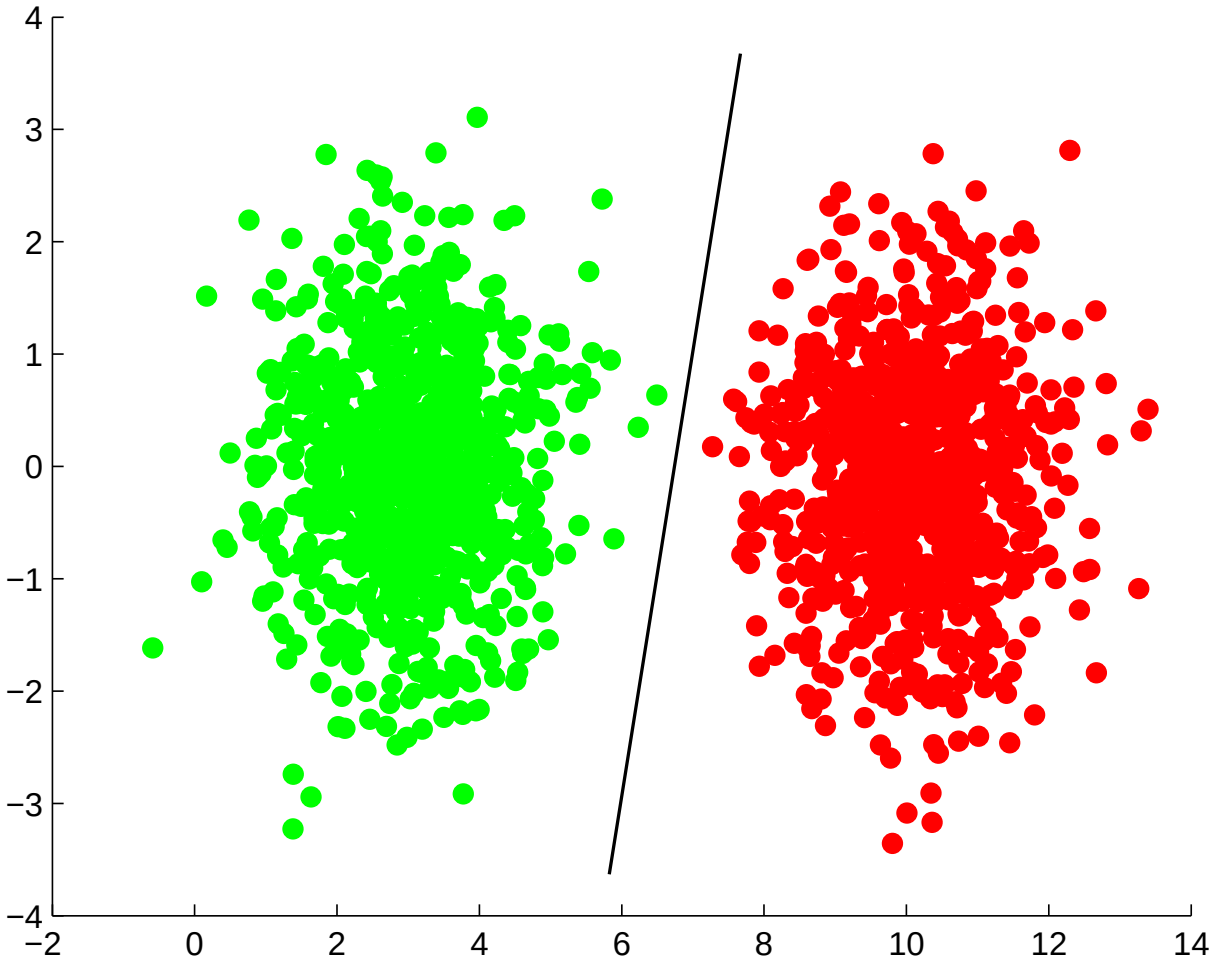


Figure 2.21: A decision surface in $\mathbb{R}^2$ space.

separator dimensionality will be of $N - 1$.

A hyperplane is defined by a point (P_0) and a perpendicular vector ($\vec{w}$) to the plane at that point. This is demonstrated in Figure 2.23. Firstly, we consider $\vec{x_0} = \vec{OP_0}$ and $\vec{x} = \vec{OP}$, where P is some arbitrary point on a hyperplane. If P is on a hyperplane, then the condition for it is that $\vec{x} - \vec{x_0}$ is perpendicular to $\vec{w}$. By the law of cosines, the dot product of any two vectors perpendicular to each others is 0, and therefore, we can derive the SVM hyperplane equation in Equation 2.29.

Therefore for any hyperplane in $\mathbb{R}^D$ space, $\vec{w}$ is a normal vector to the hyperplane which specifies the orientation of the hyperplane, and the bias b determines the offset of the hyperplane from the origin. For different values of b, we get parallel hyperplanes. The distance between two parallel hyperplanes $\vec{w} \cdot \vec{x} + b_1 = 0$ and $\vec{w} \cdot \vec{x} + b_2 = 0$ is equal to $D = |b_1 - b_2|/||\vec{w}||$.

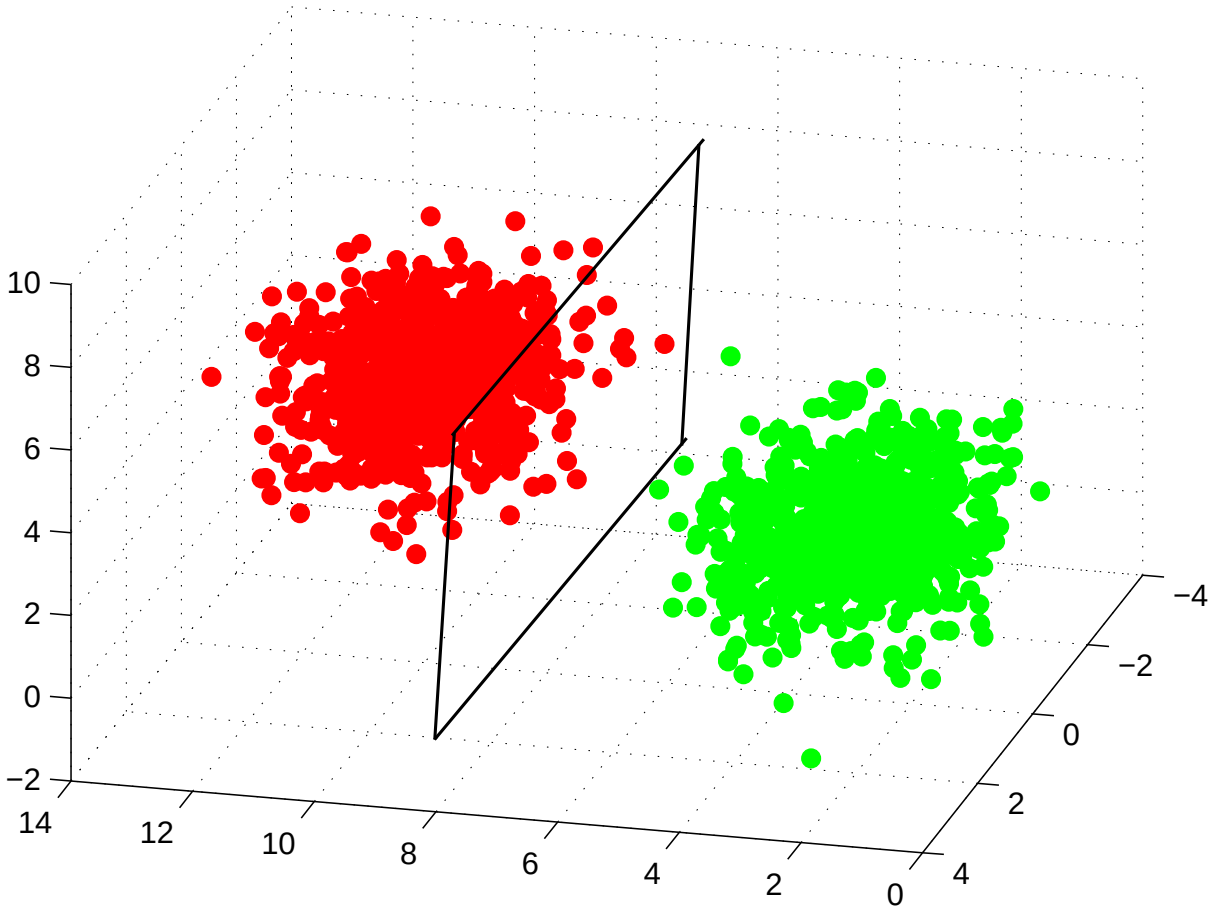


Figure 2.22: A decision surface in $\mathbb{R}^3$ space.

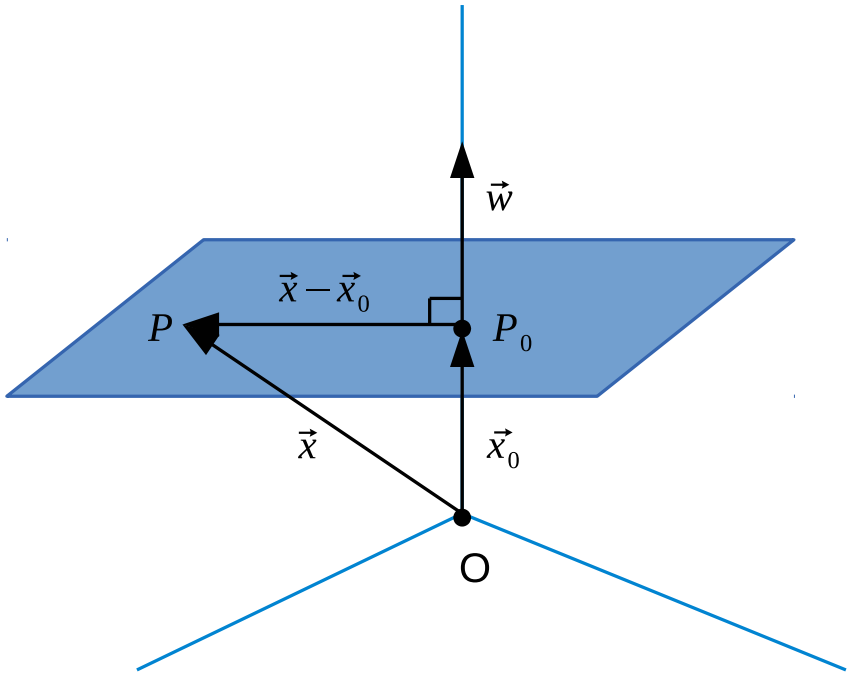


Figure 2.23: Equation of a hyperplane derivation.

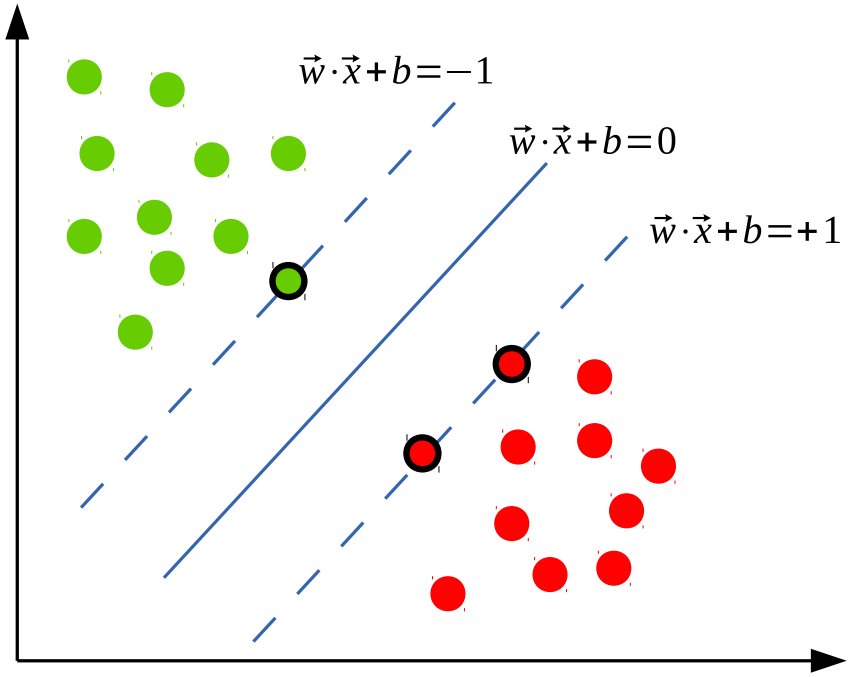


Figure 2.24: A SVM with multiple decision hyperplanes.

2.4.3.1 Primal Formulation

There are many possible hyperplanes that can separate the classes. The objective of SVM training is to find the best hyperplane [76] — the one that maximizes the distance between the hyperplane and the support vectors (samples/vectors lying on the margins across from the hyperplane). This is called the maximum margin hyperplane. An example of this is shown in Figure 2.24, where the hyperplane is maximized between two margins which are defined by the marked support vectors for the respective classes. The task of the learning stage is therefore of finding the best separation for each class reference set. We can see that the positive class (green) is defined by a support vector that lies on the hyperplane $\vec{w}\cdot\vec{x}+b=-1$. Conversely, the negative class (red) is defined by a support vector that lies on the hyperplane $\vec{w}\cdot\vec{x}+b=+1$. The position of other samples in these classes do not matter to the SVM, since we are only interested in finding a hyperplane in between the support vectors of opposing classes that maximizes distance from the support vectors to the hyperplane. This is the optimization problem that a SVM solves.

$$\begin{aligned} \vec{w}\cdot(\vec{x}-\vec{x_0}) &= 0 \\ \vec{w}\cdot\vec{x}-\vec{w}\cdot\vec{x_0} &= 0 \\ \vec{w}\cdot\vec{x}+b &= 0 \end{aligned} \tag{2.29}$$

This enables us to define linear constraints on the problem. Primarily, for the positive class, $\vec{w} \cdot \vec{x} + b <= -1$ and for the negative class, $\vec{w} \cdot \vec{x} + b >= +1$. These constraints make sure that all the samples are correctly classified. If we assume that y_i denotes the class label for a particular sample, then, the full constraint is defined in Equation 2.30.

$$\begin{aligned} \vec{w} \cdot \vec{x} + b <= -1 \text{ if } y_i = -1 \\ \vec{w} \cdot \vec{x} + b >= +1 \text{ if } y_i = +1 \\ \text{therefore} \\ y_i(\vec{w} \cdot \vec{x} + b) >= 1 \end{aligned} \tag{2.30}$$

The gap between one class and the other is defined as the distance between the parallel hyperplanes. Since $D = |b_1 - b_2|/||\vec{w}||$, then $D = 2/||\vec{w}||$, defines this distance. To maximize the gap between hyperplanes we can equivalently minimize $||\vec{w}||$ (or $\frac{1}{2}||\vec{w}||_2^2$), subject to the constraint in Equation 2.30. In the test case, a new instance $\vec{x}$ is classified by Equation 2.31.

$$f(\vec{x}) = sign(\vec{w} \cdot \vec{x} + b) \tag{2.31}$$

This derivation is called the *primal formulation* of the linear SVM optimization problem. The summary is given in Equation 2.32, for a problem with D variables $(w_i, i = 1, \ldots, D))$ where D is the number of dimensions in the dataset.

$$\text{Minimize } \frac{1}{2} \sum_{i=1}^{D} w_i^2 \text{ subject to } y_i(\vec{w} \cdot \vec{x} + b) - 1 >= 0 \text{ for } i = 1, \ldots, D \tag{2.32}$$

2.4.3.2 Dual Formulation

It is often common to reformulate the primal formulation into the *dual formulation*, with the application of Langrange multipliers. A Lagrangian is defined in Equation 2.33 for N variables $(\alpha_i, i = 1, \ldots, N)$, where N is the number of samples.

$$\Lambda_P(\vec{w}, b, \vec{\alpha}) = \frac{1}{2} \sum_{i=1}^{D} w_i^2 - \sum_{i=1}^{N} \alpha_i (y_i(\vec{w} \cdot \vec{x_i} + b) - 1) \tag{2.33}$$

It is beyond the scope of this chapter to derive the dual formulation of linear SVM optimization. The dual formulation is defined in Equation 2.34. The solution is defined in Equation 2.35.

$$\text{Mazimize } \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{N} \alpha_i \alpha_j y_i y_j \vec{x_i} \cdot \vec{x_j} \text{ subject to } \alpha_i >= 0 \text{ and } \sum_{i=1}^{N} \alpha_i y_i = 0 \tag{2.34}$$

$$f(\vec{x}) = sign(\sum_{i=1}^{N} \alpha_i y_i \vec{x_i} \cdot \vec{x} + b) \tag{2.35}$$

There are two important points to make here. The first is that the solution is now dependent only on dot products of original data, and not the data itself. Also, the number of free parameters is bounded by the number of support vectors, and not the dimensionality of the data e.g. a dataset of 100 samples with 2000 dimensions will only require 100 parameters.

In the dual form, the only variables which need to be calculated are the Lagrange multipliers of the form α_i. By finding these Lagrange multipliers it is then possible to maximize the dual form with its constraints. The solution to this quadratic optimization is in the form of quadratic programming.

2.4.3.3 The Kernel Trick

So far we have discussed the use of SVMs in cases where classes are linearly separable. In many cases, however, there will be classes that do not have a linear separation. This concept is illustrated in Figure 2.25. A linear separation to separate the positive (green) from the negative (red) class does not exist. However there does a exist a non-linear separation as shown in Figure 2.26. The original points have been transformed into a higher dimensional space, where it would be possible to find a linear hyperplane to 'slice' between the two classes. The linear hyperplane in $\mathbb{R}^3$ space translates to a non-linear separation in $\mathbb{R}^2$ space via the transformation. The convention is to use the symbol ϕ to denote the transformation.

The technique of data transformation in higher dimensional spaces makes finding class separation much easier. The problem is to find an optimal transformation ϕ to apply to training and testing data, and then performing normal linear SVM training and testing. However, this can lead to impractical computational requirements. To solve this, the 'kernel trick' is used. Based on the 'dual formulation' described earlier, it is possible to not require the explicit calculation of data samples from one subspace to another of higher dimensionality, but rather the pair-wise dot products of the data samples. There exist a set of functions called kernels that are capable of calculating the dot product in a higher dimensional subspace without explicitly transforming the original samples via ϕ. The kernel functions act in the original subspace,

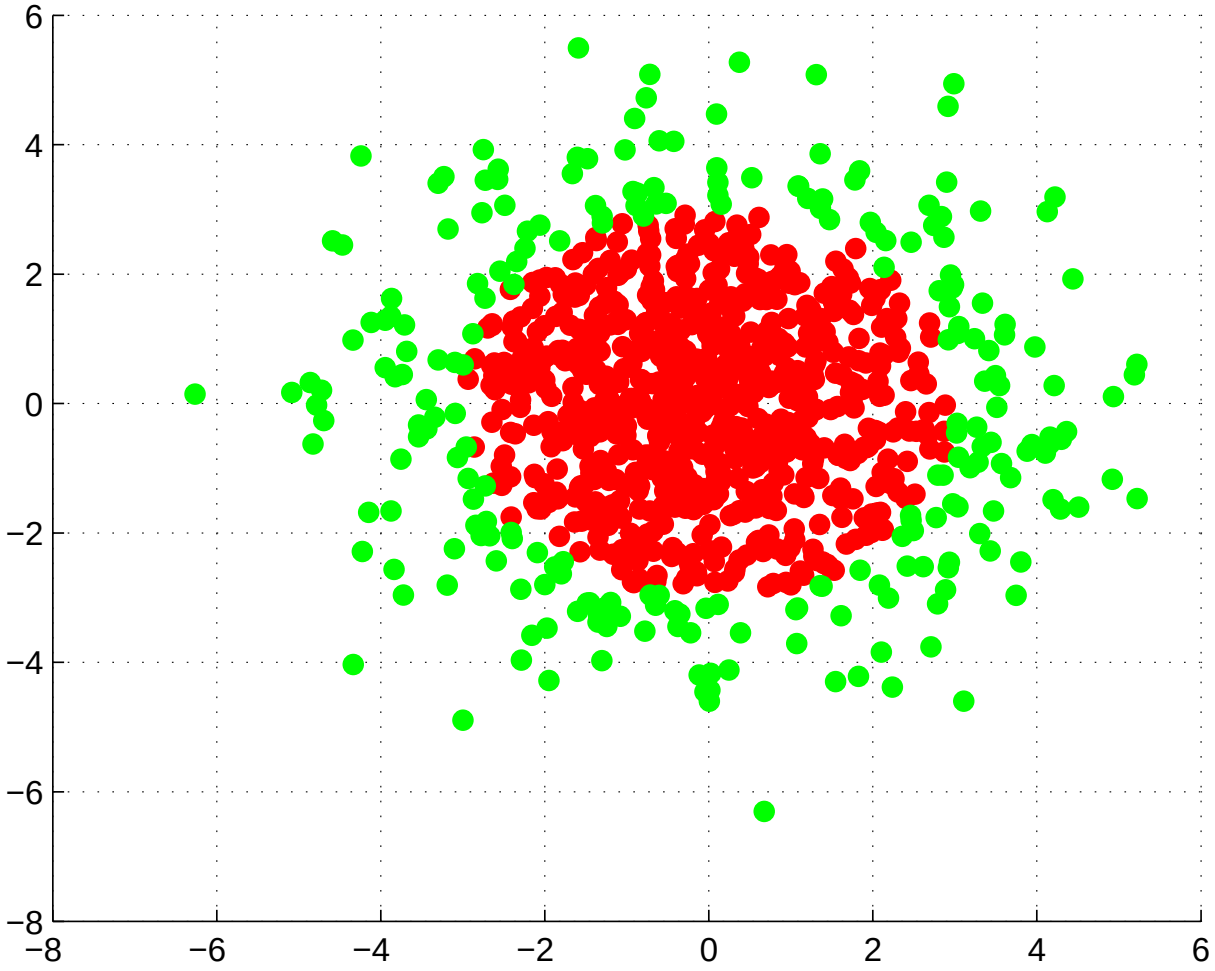


Figure 2.25: No linear decision surface in $\mathbb{R}^2$ space.

requiring no additional memory. The only addition is computational time to calculate the kernel $K(\vec{x}_i, \vec{x}_j)$ for all data pairs. Not all functions can be used as kernel functions. To be used as kernels, the functions must satisfy Mercer's conditions [80]. An example will make this clearer. Suppose we have two dimensional data: $\vec{x} = (x_1, x_2)$ and $\vec{z} = (z_1, z_2)$. We define a kernel $K(x, z) = \langle \vec{x} \cdot \vec{z} \rangle^2$. The derivation of a kernel into a transformation based on ϕ is shown in Equation 2.36.

$$\begin{aligned} K(x,z) &= \langle \vec{x} \cdot \vec{z} \rangle^2 \\ &= (x_1 z_1 + x_2 z_2)^2 \\ &= (x_1^2 z_1^2 + 2 x_1 z_1 x_2 z_2 + x_2^2 z_2^2) \\ &= \left\langle (x_1^2,\ \sqrt{2} x_1 x_2, x_2^2) \cdot (z_1^2,\ \sqrt{2} z_1 z_2, z_2^2) \right\rangle \\ &= \langle \phi(\vec{x}) \cdot \phi(\vec{z}) \rangle \end{aligned} \tag{2.36}$$

It is now apparent how the mapping function ϕ is fused within the kernel K, where

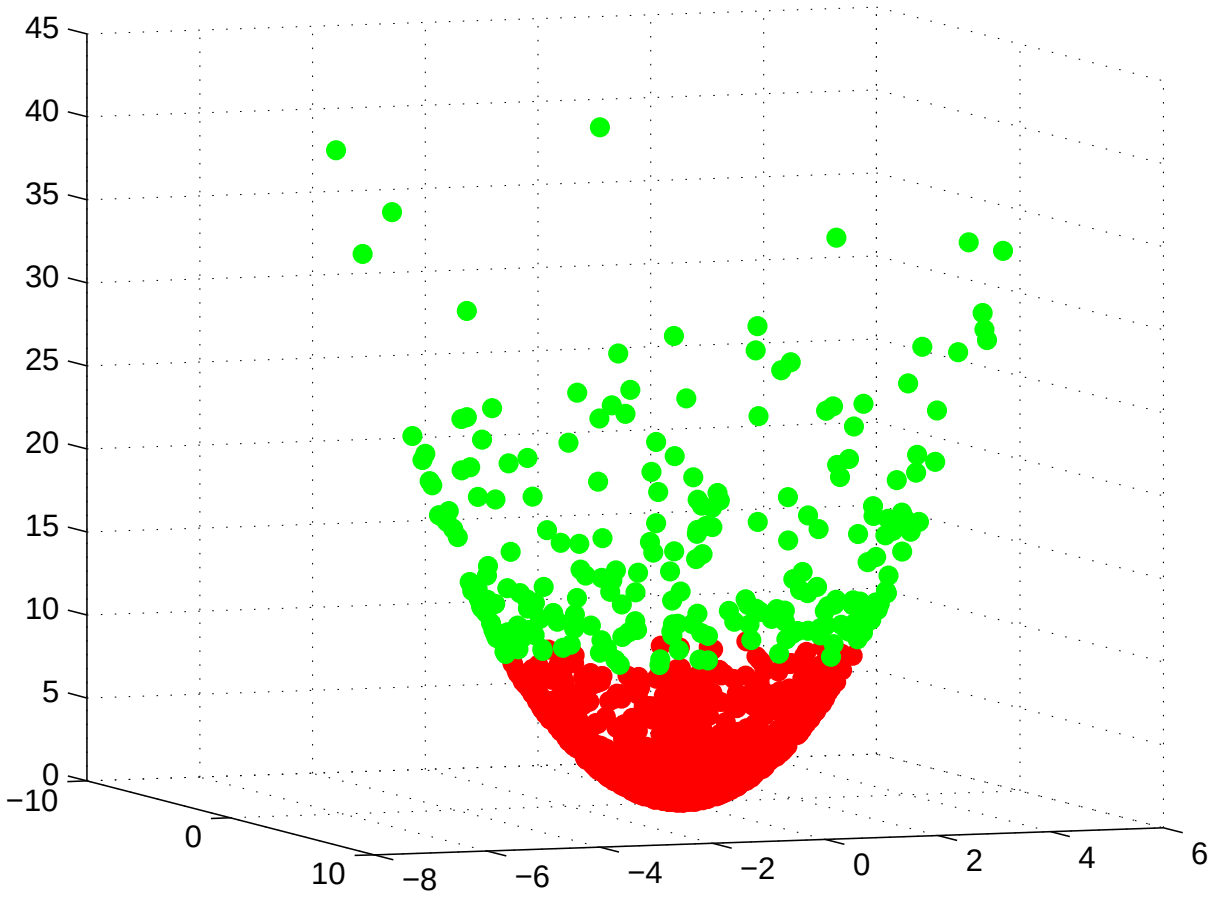


Figure 2.26: Original data plotted in $\mathbb{R}^3$ space by the transformation $[x_1, x_2] = [x_1, x_2, x_1^2 + x_2^2]$

$\phi(\vec{x}) = (x_1^2, \sqrt{2}x_1x_2, x_2^2)$. However the higher dimensional subspace produced by ϕ is no longer required. Only the dot product-based kernel form is needed. The result is equivalent in the original dimensionality subspace. Given that the data is maintained in the original subspace dimensionality, the SVM can be solved in dual form, also depending on the dot products of data pairs rather than the actual data samples in expanded subspace form. With most classes in speech problems such as GID, AID and SID represented by highly non-linear samples, SVMs have become a very important tool for classification.

2.4.3.4 Soft Margin SVM

We have so far discussed SVMs where we expect datasets to be completely separable. This is rarely the case in real datasets. To help with this situation, it is necessary to introduce slack variables. Whereas before, our margins were defined by the hard margin constraints of Equation 2.30, the new constraints with slack variables is as in Equation 2.37.

$$y_i(\vec{w} \cdot \vec{x} + b) >= 1 - \xi_i \quad \text{where} \quad \xi_i >= 0 \tag{2.37}$$

This new constraint permits a functional margin that is less than 1, and an associated penalty (or cost) often denoted by C, calculated as $C\xi_i$ for all data points that fall within the margin on the correct side of the separating hyperplane, when $0 < \xi_i <= 1$, or on the wrong side of the separating hyperplane when $\xi_i > 1$. What this aims to achieve is a margin that classifies the training data as correctly as possible whilst at the same time softening the constraints to allow for non-separable data. The penalty of misclassified points in training data is proportional to the amount of misclassifications. The value C can therefore be modified to achieve varying levels of flexibility.

2.4.4 Kernel Function and Parameter Selection

In the work presented in this thesis, we make use of the RBF kernel. This seems to be a reasonable first choice, if not optimal to the specific problem. The RBF kernel provides a nonlinear mapping of samples to a higher dimenionality space. Given that the AID problem provides classes where the relation between class labels and their attributes are nonlinear, then a nonlinear kernel is an appropriate choice. It is also possible to model a linear kernel with a RBF kernel, depending on the choice of parameters. A linear kernel with a cost parameter of C can have equivalent performance with a RBF kernel SVM with specific parameters (C,γ). The same can be said for the sigmoid kernel given certain parameters [81, 82].

Another reason why the RBF kernel is a popular choice is the number of hyperparameters that influence the behaviour of this kernel. Other kernels e.g. the polynomial kernel can be more complex to tune. The RBF kernel has also less numerical difficulties with kernel values falling between hard limits as opposed to polynomial kernels where kernel values have far less strict bounds. The RBF kernel is however not suitable in cases where the feature dimensionality is very large. The next section will in fact discuss popular dimensionality reduction methods which are often performed on the original feature space prior to utilising a RBF kernel. It is also suggested that when the feature space is of a high dimensionality, it is probably best to just use a linear kernel. This thesis does not go into very much details exploring new kernel functions, nor does it go into any details justifying which kernel choice would be most appropriate. The use of a RBF kernel is therefore meant as a heuristic, or rule-of-thumb choice, rather than founded in a geometric analysis of the feature space.

Another important factor when utilizing SVM is the selection of appropriate hyperparameters for the problem. The RBF kernel requires two hyperparameters to be configured, the cost C and the gamma value γ. There is no prior way of knowing which combination of values is best suited for any problem. One of the popular methods for determining good parameters is the grid-search approach. The idea is to split training data further into a training set and a development set. Various parameters are tried in grid-search fashion, and a RBF kernel trained on the training set, and tested on the development set. This way, the development set is an unseen portion of data, similar to future test sets. Multiple training/development folds are tried for cross-validation. During this procedure, the aim is to find a set of parameters that isn't specifically aimed at getting the highest performance for the test set (this leads to overfitting), but a set of parameters that generalize well to unseen data sets. The LIBSVM toolkit makes a suggestion of trying out different pairs of (C,γ) with exponentially growing sequences for these hyperparameters. This thesis does not perform this kind of cross-validated grid search. The main reason for this is that we feel there is not sufficient data in the ABI-1 corpus to have training sets split further to include a development data set. Insofar as grid-search parameter tuning is not performed, all SVMs used are based on their default parameters in LIBSVM. We think this fact is important to state and we will revisit this claim when surveying results for SVM-based classification.

2.5 Dimensionality Reduction

In many classification problems, it is quite common to perform some form of dimensionality reduction of the feature space prior to building a model for classification. We shall discuss two important methods: Principal Component Analysis (PCA) and Linear Discriminant Analysis (LDA), which can be used separately, or combined together.

The prime difference between LDA and PCA is that PCA does individual feature classification, in order to rank each feature by the amount of information it provides with respect to the data, and is blind to any classes present. A common analogy is that we would have more information about a pen if we look at a projection of it showing the side, rather than a projection showing the point. On the other hand, LDA does data classification with respect to discriminating classes in a supervised way. In PCA, the shape and location of the original data sets changes when transformed to a different space whereas LDA only tries to provide more class separability and draw a decision region between the given classes.

2.5.1 Principal Component Analysis

The aim of PCA [83] is to find the principal components of a dataset. The principal components are the directions of most variance in the data. The intuitive idea of PCA is to rotate the entire dataset about an axis (or plane), such that maximal variance in the data can be seen. The direction of rotation is defined by the principal components under which the majority of the variability is visible. This concept is demonstrated in Figure 2.27. When the data points are projected (via the arrows) to different principal components (p_1 and p_2), we can see that p_1 accounts for a wider variance in the dataset than the direction given by p_2, and therefore this makes the first principal component a better choice to describe the variance in the dataset.

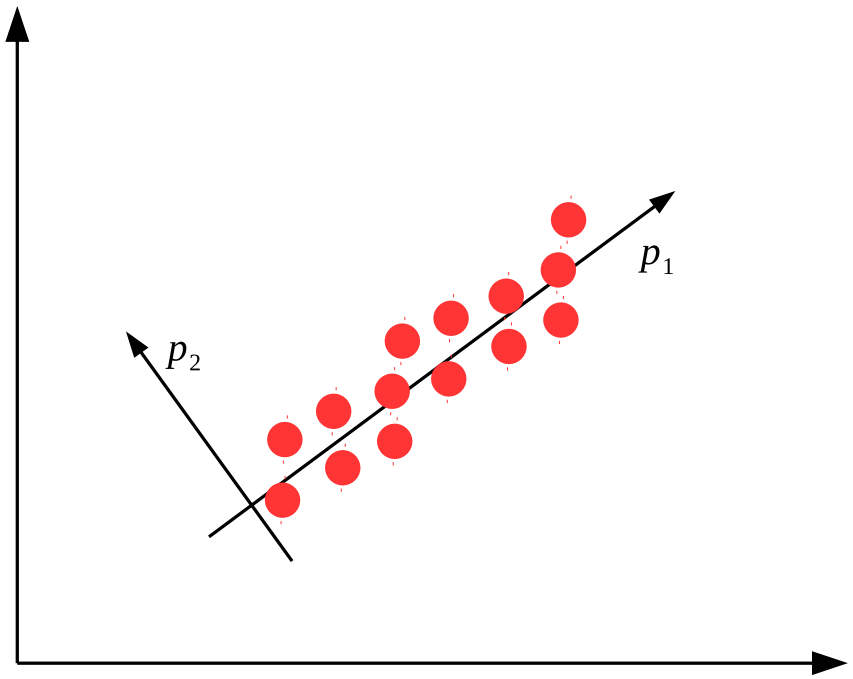


Figure 2.27: Different PCA principal components on the same dataset.

We shall define the process formally. Given a sample $X = \mathbf{x}_1, \ldots, \mathbf{x}_n, \mathbf{x}_i \in R^d$,

1. compute sample mean: $\hat{\mu} = \frac{1}{n}\sum_i (\mathbf{x}_i)$
2. compute sample covariance: $\hat{\Sigma} = \frac{1}{n}\sum_i (\mathbf{x}_i - \hat{\boldsymbol{\mu}})(\mathbf{x}_i - \hat{\boldsymbol{\mu}})^T$
3. compute eigenvalues and eigenvectors of $\hat{\Sigma}$ by Singular Value Decomposition (SVD)

An eigenvector is a direction. The eigenvectors in Figure 2.27 are the vertical or horizontal vectors to which the data points are projected. The eigenvalue is a score which gives a rank of how much variance (spread) exists in the direction of the corresponding eigenvector. The

eigenvectors are ranked by their eigenvalues. There is one eigenvector per dimension of the original dataset. Also all the eigenvectors of a dataset put together have to account for all the variance in the dataset, and therefore, all eigenvectors for a dataset are orthogonal to each other. The new axes set of the data become the actual eigenvector direction. Nothing changes in the dataset, except that it has been rotated around the eigenvector axes. Dimensionality reduction can be achieved by choosing the dimensions in the order of their maximal eigenvalues.

2.5.2 Linear Discriminant Analysis

When utilizing PCA, there is no reference to the classes within a dataset. It is unsupervised. The entire dataset is treated as a whole, and a representation of lower dimensionality is obtained whilst preserving most of the variation. LDA [84] differs in that it is a supervised technique that reduces dimensionality whilst maximizing the separability of the classes. The aim of LDA is also to find a projection for the original data, in $C - 1$ dimensions, where C is the number of classes in the data, and the ratio of *between-class* and *within-class* scatter matrix is maximized. This concept is visualized in Figure 2.28. There are two orthogonal projection lines w_1 and w_2 to which the data from the two classes can be remapped to. It is clear that a linear boundary (dashed blue) line can be maintained to distinguish the two classes if the data is projected onto w_1. On the other hand, if the data was projected onto w_2, the linear boundary between classes would not be maintained.

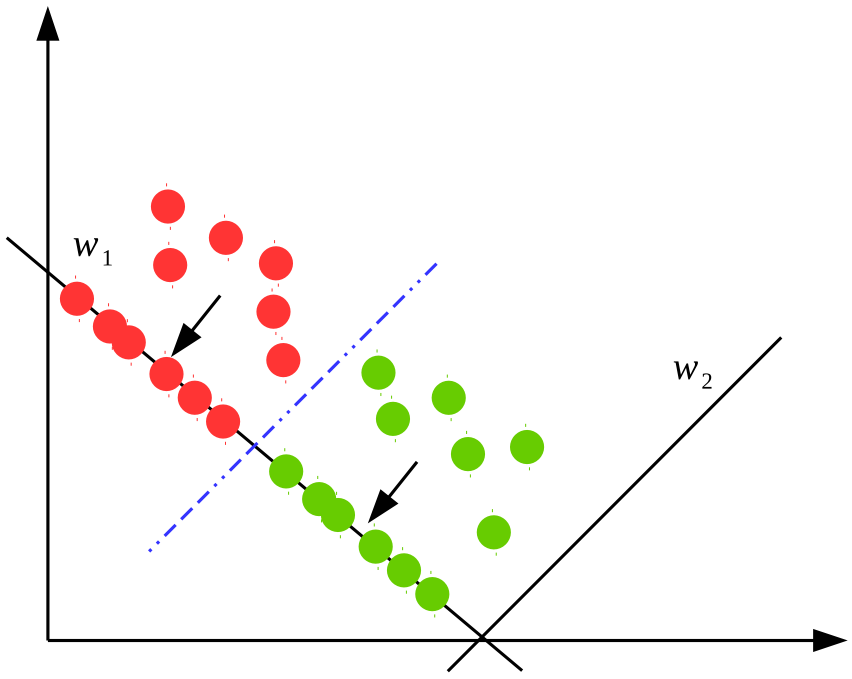


Figure 2.28: LDA maximal class separation.

The ratio between the *between-class* and *within-class* scatter matrices for LDA is defined in Equation 2.38. The term S_B is the *between-class* scatter matrix, and the term S_W is the *within-class* scatter matrix, define in Equation 2.39 and Equation 2.40 respectively. In this set of equations $\bar{\mathbf{x}}$ denotes the mean of the dataset, whilst μ_c denotes the mean of a class.

$$J(w) = \frac{w^T S_B w}{w^T S_W w} \tag{2.38}$$

$$S_B = \sum_c (\boldsymbol{\mu}_c - \bar{\mathbf{x}})(\boldsymbol{\mu}_c - \bar{\mathbf{x}})^T \tag{2.39}$$

$$S_W = \sum_c \sum_{i \in c} (\mathbf{x}_i - \boldsymbol{\mu}_c)(\mathbf{x}_i - \boldsymbol{\mu}_c)^T \tag{2.40}$$

With these matrices, it is possible to solve the maximization of $J(w)$. The eigenvectors are ranked by their eigenvalues, and the first $C - 1$ eigenvectors of the LDA projection matrix are kept. This projection matrix is then applied to the dataset to obtain the LDA-reduced dataset.

2.6 Genetic Algorithms

In this thesis, we make use of Genetic Algorithms for classifier fusion. Considering that we may employ different classifiers to classify the accent of an utterance, fusion allows us to obtain a final classification based on the combined guesses of different classifiers. There is a question as to which classifiers are better than others (and we can measure this as classification error for each classifier). Another question is that of which classifiers should be fused together to produce an optimal (or quasi-optimal) final classification, with the lowest error. This is a problem for which a GA can be employed. A GA is a search algorithm that is loosely based on Darwinian evolutionary theory and the principle of survival of the fittest, and introduced in [85]. The general template of a GA is an iterative technique where each iteration is called a "generation". In each of these generations, a number of solutions, termed "chromosomes" are evaluated, "mutated" and some form of "crossover" reproduces new "chromosomes" for the next generation. We shall go over this process briefly.

The first step to implementing a GA is to decide on the encoding of a chromosome i.e. the representation of a solution. Since we want to discover an optimal combination of classifiers for which the classification error is reduced, then we can opt for a binary encoding. Each chromosome is therefore a string of binary values (1 or 0). The value 1 indicates that the classifier should be included in the fusion, whilst 0 indicates that the classifiers should not be included in

fusion. The encoding really depends on the kind of search problem at hand. The second stage is the random set of a population of chromosomes to be the first generation of possible solutions. Assuming that we need to select the best combination out of a set of ten classifiers, then each chromosome is a binary string of ten characters. Each chromosome is assigned a fitness value, which is determined by a fitness function. The scope of the fitness function is to determine how well the chromosome can survive the scenario it is being tested for. In the case of the fusion problem, the classifiers selected by the chromosome are fused together to give a final error rate, and this is an indication of how good or bad the solution given by the chromosome is [86].

The third stage is the selection of which chromosomes are to take on the role of parents in crossover (creating new offspring). By evolutionary theory, the fittest individuals should survive more generations and take part in more creation of offspring, whilst the weaker solutions eventually die off earlier. There are many possible methods of selection. For the purposes of this thesis, the Stochastic Universal Sampling method is used [87], where a number of individuals from a shuffled order of chromosomes are selected, giving strong and weak solutions a fair chance to participate in creating other offspring. There are many available sampling methods, and this choice is mostly arbitrary. However, one of the best properties of this sampling method is that it gives weaker members of the population a chance to be chosen, reducing the bias of unfair fitness-proportional selection methods. We do this because the fitness criteria we employ may not be the absolute best for the particular problem. The fourth step is to perform crossover. There are many types of methods for crossover. For the purposes of this thesis, single point crossover is considered. This is best demonstrated by an example. Consider two parents (Chromosome A = 10011|101011 and Chromosome B = 11001|010100), where '|' is the crossover point. The segments to the right of this point are exchanged to create new children (Child A = 10011|010100 and Child B = 11001|101011) [86].

Once all crossovers are completed, the next step is mutation. Mutation is the process of a slight alteration to the information in each chromosome, say the inversion of one or more randomly selected bits. This allows for some additional diversity to the population, and helps the GA not to get locked up in locally optimal solutions, but instead, varies the gene pool for the next generation. Crossover and mutation are controlled by parameters called the crossover and mutation rates. The crossover rate is a value from 0% to 100%, and this determines how many times crossover is carried out in the current generation. Therefore, 0% means the next generation will be made entirely of the same chromosomes in the current generation, whilst 100% means that all the chromosomes in the next generation will be newly created offspring

resulting from crossover. The mutation rate (also between 0% and 100%) determines how many genes in a population are to be mutated. Each gene is one bit from each chromosome [86].

Once the crossover and mutation is complete, the new generation can start. At the start of the generation, the least fit chromosomes are removed. The number of chromosomes to keep is determined by the generation gap, which is also a value between 0% and 100%, and determines how much of the new generation is replaced. Termination of the GA is based on a predetermined number of generations or else it can be based on some convergence criteria. If the error of the best solution does not improve and is constant for a long number of generations, the GA is allowed to terminate [86].

2.7 Summary

In this chapter we have given an extensive overview of a number of concepts that are involved in acoustic classification problems. We have introduced concepts of the speech production mechanism and seen what kind of processing is performed on speech signals to extract important features. We have also seen how more meta-information can be derived from raw features, as well as what steps must be taken to remove the effect of adverse and varying recording conditions. We have gone over a number of standard classification methods which are applied to these features, and the dimensionality reduction techniques that can be employed. Finally we have given a brief description of genetic algorithms and how they can operate for finding quasi-optimal solutions for classifier fusion. There are a number of problem-specific techniques that are used in the specific areas of GID, AID, LID, SID etc. which build upon the techniques described here. We describe these in the next chapter, which relies on the background we have described here.

Chapter 3

Literature Review

In this chapter, we first give an overview of literature about human and animal speech perception theories, some of which strengthens the validity of acoustic-only models for apparently linguistic-oriented constructs such as languages. We will describe a number of classification systems that are used in acoustic classification problems. In particular, we will discuss methods that are not very specific to accent or gender classification. In fact, most systems in literature deal with speaker and language classification in general. It is only very recently that such systems have been developed for use with accents, and this thesis will ultimately deal with the state-of-the-art in accent classification.

For the purposes of this thesis, we deal only with acoustic-only classification. We shall not be discussing supervised, transcription based systems. This thesis does not make use of phonotactic systems either. However, we discuss this particular method as being somewhere in between supervised and unsupervised systems, and it is a good system to compare our accent classification results with in later chapters.

Since this thesis deals primarily with two different classification problems, GID and AID, we will be using the term "classes" as well as "languages", "accents", "genders" etc. somewhat interchangeably. We do not refer to any methods as being solely designed for one particular classification problem. We shall highlight this in the sections when this is not applicable, and when referring to specific works in literature. The first sections will deal with the construction of specific classification methods that are employed in general for the use of AID, LID, SID etc. whilst the second part of the chapter will give an overview of specific work done in the fields of GID and AID prior to this thesis.

3.1 Human and Animal Speech Perception

The work in this thesis is somewhat inspired by some details in the theory of speech perception in humans and animals. We therefore consider it an important aspect of the approach to the work presented. This section will give an overview of the major theories of speech perception in the literature, and experimental findings that motivate the acoustic-only classification systems we present.

The task of listening to and learning specific voice characteristics is crucial to classification methods from speech. We have discussed how speech is perceived in terms of an acoustic signal, which is built up of concatenated sounds (phonemes). The human ear is capable of receiving and interpreting this acoustic signal, but not before having been properly trained through experience. This is what makes an analysis of the topic of speech perception interesting for the application to a machine learning-oriented treatment of speech signals.

Phonemes are the basic units that enable us to differ between words that have similar sounds, such as 'lap' and 'rap' [88]. Phonemes are pronounced differently depending on how they are combined with other phonemes. If we had to record the words 'ran' and 'run', and isolate the /r/ phoneme, we would be able to distinguish the /r/ phonemes. This is because the articulatory configuration to produce these words, blends the phonemes, making each roughly separable acoustically, but it is still hard to determine where the shift from one phoneme to the next is occurring within the signal. This phenomenon is called 'co-articulation', and the basic unit of speech perception is co-articulated phonemes [89].

In [90], a study considers the uncertainty of whether co-articulated phonemes are the basic unit of speech that speech perception theories should build upon. In this study, humans are compared to a typewriter, which makes use of distinct phonemes. It is argued that humans do not have a separate vocal tract for each phoneme, in the way a typewriter has a separate hammer to produce each letter. Instead, our single vocal tract has to alter its shape to produce each sound. The studies of co-articulation gave rise to different theories on speech perception.

3.1.1 Motor vs. Auditory Speech Perception

A theory of speech perception must explain how humans can hear the word 'ran' and realize that the phoneme /r/ was heard. Co-articulation effects cause the phoneme /r/ to be pronounced differently with different blendings, but at the same time, seem like an indistinguishable sound

merged with another sound. The motor theory assumes that a human listener is capable of distinguishing phonemes because he himself is a speaker. The listener knows what articulatory gestures are required to produce a syllable prior to hearing it. What the speaker produces, is a coded sequence of phonemes that the listener can decode [91].

The motor theory considers speech perception to be an innate, species-specific ability. Research found that humans have special neural detectors that respond uniquely to human acoustic signals. These findings support the motor theory, in that they suggest that a physical trait evolved to support the processing of human acoustic signals [88, 91].

Another study [92] also credits the motor theory. It investigated verbal transformation effects (VTE). VTEs are the perceptual changes that occur when a waveform for a word, is repeated in a loop to a listener for a prolonged period. For example, after hearing the word 'pace' for three minutes, subjects reported hearing words that are phonologically similar, such as 'face', 'space', 'base' or 'case'. This kind of result cannot be explained in terms of how the ear functions, since all ears were fed with the same acoustic signal over and over, with no changes, but they could be explainable with motor theory as 'decoding gone wrong' in the process of repetition.

Another effect that supports motor perception theory is that of categorical perception. This concept stems from the studies related to voice onset time, which is the delay between the time the speaker opens their mouth to release air, and the time the vocal cords begin to vibrate. Studies showed how different phonemes could be recognized and categorized within specific ranges of voice onset times [88].

There have been additional studies that are somewhat related to the motor theory of speech production. In [93] a commentary is made on proposals made about disorders of cerebellar development which could be part of the cause of impairments in reading and writing — classical characteristics of dyslexia. The authors state that the ideas of this hypothesis are in agreement with the general premise of the motor theory of speech perception. Similarly in [94] we can find an investigation of different cerebrum activity as a reaction to speech and non-speech stimuli, suggesting a motor construct specifically targeted at human voice signals.

However, as research progressed, evidence was found that contradicted motor theory. Research into how animals communicate gave rise to the auditory theory of speech perception, challenging the motor theory. Chinchillas have an auditory system that demonstrates categorical perception [88]. When sounds with manipulated voice onset times were played to humans, the differences people noticed were forgotten once the sounds were categorized together [88]. Also,

categorical perception was shown to be caused by rapid decay of auditory memory, and not as a special trait of human speech [95]. Other animals were found to possess neural detectors for specific acoustic signals for each of their own species [91].

The auditory theory states that the speech perception process is not limited to humans alone, and it does not occur because of special knowledge of how speech is produced by speaker and listener. It is instead derived from general properties of the auditory system [88]. As testimony to this, in [96] studies were conducted to compare human newborns and tamarin monkeys, lending further support to the auditory theory of speech perception. The study showed how human newborns and tamarins were both capable of discriminating sentences from Dutch and Japanese. The study setup was the same for both humans and monkeys, with the exception of how speech perception was exhibited. The monkeys were habituated with a particular language. Then upon hearing the same language again, after utterances from another language, the monkeys tilted their heads towards the loudspeaker in recognition. No special training was required for this study. Both monkeys and humans noticed changes in language and even speaker. The authors suggested that this experiment was enough to show that humans are simply relying on general properties of the primate auditory system, in common with the tamarins. However this is not enough to explain, in totality, some speech perception phenomena.

3.1.2 Parallel vs. Serial vs. Active Speech Perception

We shall now try to look at models of speech perception that are targeted to explain how the perception process works, in terms of sequences (rather than physiology). These sequences can be divided into two main perspectives; series models with a sequential order to each sub-process, or parallel models with several sub-processes acting simultaneously. We shall also mention active models which refer to an active listener who generates an internal by-signal to assist the speech perception process.

A series model begins with the listener receiving the speech signal, and then subjecting it to auditory analysis. Phonetic analysis then passes the signal to a morphological (or lexical) analysis block. Finally a syntactic (or grammar) analysis of the signal results in a semantic (or meaning) analysis of the message. Each block in the series is said to reduce/refine the signal and pass additional meta-parameters to the next level for further processing [97]. Series models imply that decisions made at one level of processing affect the next level, but do not receive feedback. In [97] it is suggested that speech perception is, in reality, more dynamic, and that series models are not an adequate representation of the process.

The need to extract several phonemes from any syllable at once is taken as evidence that speech perception is a parallel process. Parallel models demonstrate simultaneous activity in the sub-processes of speech perception. This means that a process at one level may induce a process at another, without any sequential hierarchy [97]. The parallel model is made up of five successive stages:

1. acoustic parameter extraction
2. micro-segment detection
3. identification of phonetic elements
4. identification of sentence structure
5. semantic interpretation

Each stage includes meta-data of previously acquired knowledge, though not necessarily in the previous stage of processing. The system also has comparator modules between the successive stages, with the possibility of direct connection between the lowest and highest levels [97]. Work in [94] is in agreement with this and makes an important conclusion about speech and non-speech analysis, in that the results suggested early separation of speech and non-speech auditory processing. Humans process speech differently at a very early stage of the perception process, and a full acoustic analysis is not performed before it is processed as speech, as the series theory would suggest.

In [98] the effects of varied speaking rate on perception were investigated, suggesting the active models of speech perception. When we speak, we may change our speaking rate throughout the dialogue. An example demonstrated in [98] is the problem of differentiating a /w/ at a fast speaking rate (which may be a /b/ at a slower rate). Listeners must have a notion of rate to be able to accurately grasp the speaker's message. Humans therefore are capable of normalizing rate differences in real-time. This is consistent with views of active cognitive systems, and normalization is an actively controlled process. To test this theory, participants in an experiment were given the task of locating a target phoneme as quickly as possible amongst 16 syllables read at two different rates. The participants were split up in two groups. The first group was read words at a constant rate, whilst the second group was read words at a varied rate. The recognition times of the first group were faster than those of the second group. This result is consistent with the hypothesis that rate normalization does increase the cognitive load of the listener, and is active in the perception process.

3.1.3 Multimodal Speech Perception

We have so far spoken about speech perception in terms of acoustic signals. However, an interesting perspective of human speech perception is that it is multimodal in nature. The McGurk effect [99] was a result of research that linked acoustic signal recognition to processing of facial expressions corresponding to the speech waveform. It is possible to confuse speech perception by showing a video of a person mouthing one phrase, whilst simultaneously playing another phrase on a loudspeaker.

In [100] more studies examined the McGurk effect on children. These results were important because they determined to which extent our visual signal integration is learnt with age, or whether this effect is innate. The studies showed how changes of the McGurk effect are detectable as we age. It appears that there are differences in the knowledge of how aural and visual data are used in speech perception depending on the age group. The implication here is a question of what metric is best for combining aural and gestural cues. In [101] evidence is used to show that word recognition accuracy using two sources is greater than the sum of the individual sources.

Another interesting study supporting multimodal theories of speech perception is that of speaker normalization (as an active process) in the context of speaker gender. The conclusion is that speaker normalization is based on abstract, subjective knowledge of the conversation. Speakers modify their perception according to the totality of information available, including direct cues from gender, which would imply different pitch in voice, and different visual cues. When listeners identify a talker as either male or female, they automatically set expectations for what the talker should sound like, and employ these expectations actively for speech perception [102].

3.2 Phonotactic Systems

The state-of-the-art approach to LID has been based on phonotactic systems [103, 104, 105, 106, 107, 108, 4, 109, 110]. The idea in these systems is to start off from the fact that there are very observable language-dependent differences in the sequences of sounds between one language and another. By modelling these sequences explicitly, we are able to build a Language Model (LM) for each language, which differ considerably from others.

If we consider the fact that words of any language are made up from a sequence of phonemes, then it stands to reason that although there is great overlap in the set of phonemes across

languages, the vocabulary of different languages differs, and as a consequence the observed sequences of phonemes differs as well. We can think of AID in a similar fashion. The language is the same, so we do not require a LM that tracks phoneme sequence differences across languages. However, we have defined accents earlier as the different pronunciation of the same word, from one region to another. The changes in pronunciation result from a change in the realization of phonemes for a specific word or phrase, in particular accent regions. So though the extent of differences in accents are somewhat less obvious than those across language, we can still build a LM for different accent regions. Put simply, accents are treated as different languages. The construction of the phonotactic system is unchanged.

3.2.1 Phone Recognition

The first stage of a phonotactic system is phone recognition - sometimes called tokenization. In this stage, the speech signal has to be converted into a sequence of symbols (or tokens), each representing a particular phoneme. In order to do this, a phone recognizer is used. There are two approaches to performing tokenization. The most common method used is based on phoneme recognition from triphone trained models (one of the blocks in an ASR system), whilst the second method is based on a GMM acoustic space model. We refer to these as supervised and unsupervised phoneme recognition respectively.

3.2.1.1 Supervised Phoneme Recognition

In the case of LID, since many languages are involved, the phone recognizer is preferably language independent. If the phone recognizer is language-dependent, then the phone recognizer would have been limited in training by the phones that exist in that particular language. There are some phones that are specific to one language and not another. This information is important for LID, and therefore it is preferable to have multiple phone recognition systems trained on different languages, working together, to cover a wider set of possible phones. The advantages here are two-fold. Firstly, a phone recognizer trained on one language, will produce a language specific error for unobserved phones coming from other languages. Secondly, the phones of a particular language that are not in another are catered for by at least one of the phone recognizers, weighting up these observations when compared to others common to all languages. The conventional LID systems based on a phone recognizer trained on a single language are referred to as Phone Recognition Language Modelling (PRLM), whilst those built on multiple language

phone recognizers are in turn called Parallel Phone Recognition Language Modelling (PPRLM). This kind of phoneme recognition system is one of the building blocks of a typical ASR system, and a typical configuration is a phone decision tree triphone recognizer, using MFCC feature vectors, where every state is represented by a low-order GMM e.g. 8 components. This is combined with a dictionary of triphones and words of a particular language. Therefore in order to build such an LID system, there needs to be a phonetic transcription of training data to build a phone recognizer — this is seen as a limitation for adding additional languages, since each language is often built with word-level transcriptions.

3.2.1.2 Unsupervised Phoneme Recognition

There is an alternative to using an ASR frontend for tokenization. This is however a less popular choice. In this system, a GMM is built for a large amount of training data from multiple languages. We have discussed earlier how the number of components in a GMM is loosely tied to the number of phonemes of a language. Given that we are cocnerned with modelling differences across multiple languages or accents, resolution is an important factor. The more training data available, the higher the resolution (in terms of number of components) of a GMM can be. Higher order GMMs e.g. 2048 components, would provide sufficient resolution for LID or AID. Hence phoneme sequences are replaced by a sequence of GMM indexes. Every feature vector in a stream is replaced by the GMM component index which gives the highest likelihood for that vector. Once the stream of input vectors is converted to GMM indexes, the same principles for all phonotactic systems apply [44, 111].

3.2.2 Vectorization

Given phoneme sequences or tokens, it is now possible to calculate the probability of a certain sequence occurring for a particular language or accent. These sequences are called n-grams, where n represents the number of tokens in a sequence. We can set D as the set or predefined n-gram sequences that we want to observe in all utterances. Given D, every utterance can be summarized as a D-dimensional vector $p = (p_1, p_2, \ldots, p_D)$. Each term p_i refers to the probability of an n-gram C_i being observed in the utterance. The maximum likelihood estimation of p_i is defined in Equation 3.1.

$$p_i = \frac{\text{Count}(C_i)}{\sum_{j=1}^{D} \text{Count}(C_j)} \tag{3.1}$$

The result of the vectorization process is that every utterance is converted to a fixed-length vector of D dimensions. The values estimated by p_i can be weighted to emphasize class-specific components of the vector. Typical methods used are the Inverse Document Frequency (IDF) and the Log-Likelihood Ratio (LLR) [112]. Based on the results in [113], the LLR weighting method outperformed IDF for AID. The LLR weighting emphasizes the most discriminative components (n-grams that are common in one language or accent, but not in any other), whilst at the same time weighing down n-grams that are common across all the classes. The LLR weighting is shown in Equation 3.2, where g_i is a smoothing function to compress the dynamic range e.g. $g_i = \sqrt{x}$. The denominator $p(C_i|\text{All})$ is the probability of n-gram C_i across all the languages or accents.

$$w_i = g_i\left(\frac{1}{p(C_i|\text{All})}\right) \tag{3.2}$$

In the case of n-gram components that are non-existent across all accent training material, these are removed and not considered further. This reduces the dimensionality of the resulting feature vectors. This is especially useful for higher order n-gram models, where empty components are quite common, and the dimensionality (due to more possibilities) is generally very large, increasing exponentially as S^n, where S is the number of tokens or phonemes used. There is another problem to deal with when modelling n-gram systems. By Zipf's Law [114], when the term frequency for a document is ranked, the frequencies follow a decaying exponential. At the top, the high ranking words with high probability are not useful for discrimination, since they appear across all document vectors. On the other hand, the low-rank words, though seemingly very discriminatory, are unreliable and not statistically significant. The area of interest is somewhere in between, where the results collected from training data are both statistically significant and provide a range of discriminatory power. For these reasons, two thresholds can be defined. First, the threshold $T1$ is a maximum value on the weightings w_j, so that no minority of components dominates the scores alone. Another threshold $T2$ is a minimum value on the weightings, such that the most common components are de-emphasised further.

3.2.3 SVM Language Model

In a traditional LID system, the n-grams gathered over an utterance could be used to train language specific n-gram language models. The set of possible n-grams would be equivalent for all languages (or accents). The LMs are trained with the maximum likelihood criterion. The

n-gram is translated to a conditional probability value that is based on the current phone given the preceding $n-1$ phones. This can be done by collecting counts of n-grams from training data, successively increasing n to collect higher order counts and using the history of phones to calculate the conditional probabilities of higher order n-grams. In [115] it is demonstrated that n-gram statistics can be computed from the n-gram posterior probabilities taken from the phone lattices generated by a phone recognizer. The best lattices for a phone sequence observation are all taken into account, and therefore the estimates of the n-gram probabilities are more reliable, improving performance. This concept is usually described as a limited context 'chain rule' of n-gram probabilities. The algebraic chain rule would consider the entire (and not immediate) context in calculating the likelihood of a sequence. If we consider a sequence of components occurring after each other as $P(C_1C_2 \ldots C_3)$, we can define a context-limited 'chain rule' as in Equation 3.3.

$$P(C_1C_2 \ldots C_3) = P(C_1) \times P(C_2|C_1) \times \ldots \times P(C_i|C_{i-1}) \tag{3.3}$$

An alternative to this is to use SVMs to discriminatively train n-gram models. The weighted vectors described above are used as a feature vector for SVM training and testing. This has proven very effective for LID, and actually outperforms the traditional n-gram based LM technique. One SVM is trained for every class, discriminating it from all other classes. This concept can also be extended for the PPRLM system. Multiple systems using phone recognition trained on different languages are constructed, and the SVM LMs for each are utilized in parallel, and scores are fused together for a final classification. An overview of a SVM LM-based phonotactic LID system is given in Figure 3.1.

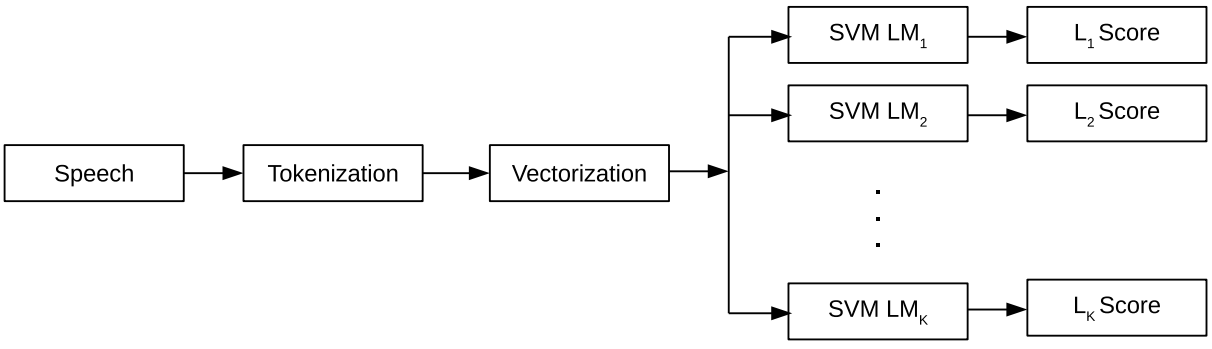


Figure 3.1: A block diagram of a phonotactic LID system.

3.3 Acoustic Systems

Another way of modelling acoustic classes as opposed to one based on phonotactics, is to use acoustic modelling. The idea here is to use class-specific features to build a model for each class. The features of the different languages should be different enough so that the contructed acoustic model will be able to model these differences. The resulting language-dependent model (or accent-dependent) models can then be used to perform classification on test feature vectors. The idea of acoustic systems is very popular in SID, since the short-term spectrum of speech contains a lot of speaker-specific information, and therefore the resulting acoustic models perform very well. The approach is less popular in LID systems, though some positive results have been achieved in literature. Acoustic systems can be split into systems based on generative models (GMMs) [116, 117, 118, 119, 120], and others based on discriminative models e.g. (SVMs) [116, 45, 120, 121, 122].

3.3.1 GMM-UBM Classifcation

In a GMM-UBM classification, a UBM is constructed from front-end feature from all languages (or accents, speakers etc.). Class-dependent GMMs are constructed by MAP-adptation of the UBM to training data for the particular class. There is a consensus that only means-adaptation is usually required, and the covariances are not updated in the process. The standard way of using a GMM for classification is to have a test utterance converted to the same feature vectors, and the GMM class model giving the best likelihood for the features of an utterance identifies the utterance as being from that class.

3.3.1.1 Kullback-Leibler Divergence

In the case of class verification rather than identification from a pool of possible classes the Kullback-Leibler divergence [123] can be used to measure the 'distance' between two GMMs. Given two probability distribution models $f(x)$ and $g(x)$, which are models representing two classes f and g, the Kullback-Leibler divergence is defined as in Equation 3.4 [124].

$$KL(f\|g) = \int f(x) \log \frac{f(x)}{g(x)} \, \mathrm{d}x \tag{3.4}$$

This divergence measure satisfies three properties:

1. Self similarity: $KL(f||f) = 0$.

2. Self identification: $KL(f||g) = 0$ only if $f = g$.

3. Positivity: $KL(f||g) \geq 0$ for all f, g.

The divergence measure however, is not symmetric. A variant that provides symmetry is more commonly used for measure similarity in classes in speech applications, and is defined as in Equation 3.5.

$$KL_{\text{symmetric}}(f||g) = KL(f||g) + KL(g||f) \tag{3.5}$$

When data is used to MAP-adapt a UBM to create a class or utterance-specific GMM, the Kullback-Leibler divergence for two GMMs is bounded by Equation 3.6 [125].

$$KL_{\text{gmm}}(f||g) \leq KL(w^f||w^g) + \sum_{1}^{M} w_i^f KL(\mathcal{N}(.;\boldsymbol{\mu}_i^f, \Sigma_i^f)||\mathcal{N}(.;\boldsymbol{\mu}_i^g, \Sigma_i^g)) \tag{3.6}$$

The term $KL(w^f||w^g)$ is the Kullback-Leibler divergence between the weights w^f and w^g. The term $KL(\mathcal{N}(.;\boldsymbol{\mu}_i^f, \Sigma_i^f)||\mathcal{N}(.;\boldsymbol{\mu}_i^g, \Sigma_i^g))$ is the Kullback-Leibler divergence between the i^{th} Gaussian component of the GMM f and the i^{th} Gaussian component of the GMM g. For this bound to be correct the i^{th} Gaussian components must correspond to each other, which is usually the case when a GMM component is the MAP-adapted component of a UBM.

Given that we have said earlier that in most speech applications such as SID and LID, only the means of a UBM are adapted to form a GMM, the symmetric Kullback-Leibler divergence for two GMMs can be simplified as in Equation 3.7.

$$\begin{aligned} KL_{\text{gmm}}(f||g) &= KL(f||g) + KL(g||f) \\ &= \sum_{i=1}^{M} w_i \left(\boldsymbol{\mu}_i^f - \boldsymbol{\mu}_i^g\right)^T \Sigma_i^{-1} \left(\boldsymbol{\mu}_i^f - \boldsymbol{\mu}_i^g\right) \\ &= D(f, g) \end{aligned} \tag{3.7}$$

The term $D(f, g)$ gives a similarity measure valid for a means-only adapted pair of supervectors for GMMs f and gIt serves as an upper bound for the Kullback-Leibler divergence. If two GMMs are far from each other, then $D(f, g)$ for the two GMMs will be large, and the converse is also true.

Recalling GMM theory, we can define a score for a particular utterance $X = \{x_1, x_2, \ldots, x_N\}$

as belonging to a particular GMM m with Equation 3.8, where λ_{ubm} represents the UBM.

$$\text{Score}_m(X) = \frac{1}{N}\sum_{n=1}^{N} log\left(\frac{P(x_n|\lambda_m)}{P(x_n|\lambda_{\text{ubm}})}\right) \tag{3.8}$$

With this in mind, SID or LID (or any other classification) can be scored equivalently via the Kullback-Leibler divergence between the GMM model m and the UBM model as in Equation 3.9 [124], by utilizing terms of the form in Equation 3.7, where λ_x is the MAP-adapted GMM obtained from the test data.

$$\text{Score}_m(X) = D(\lambda_x, \lambda_{\text{ubm}}) - D(\lambda_x, \lambda_m) \tag{3.9}$$

A block-diagram overview of the verification process of an utterance belonging to a particular class is given in Figure 3.2.

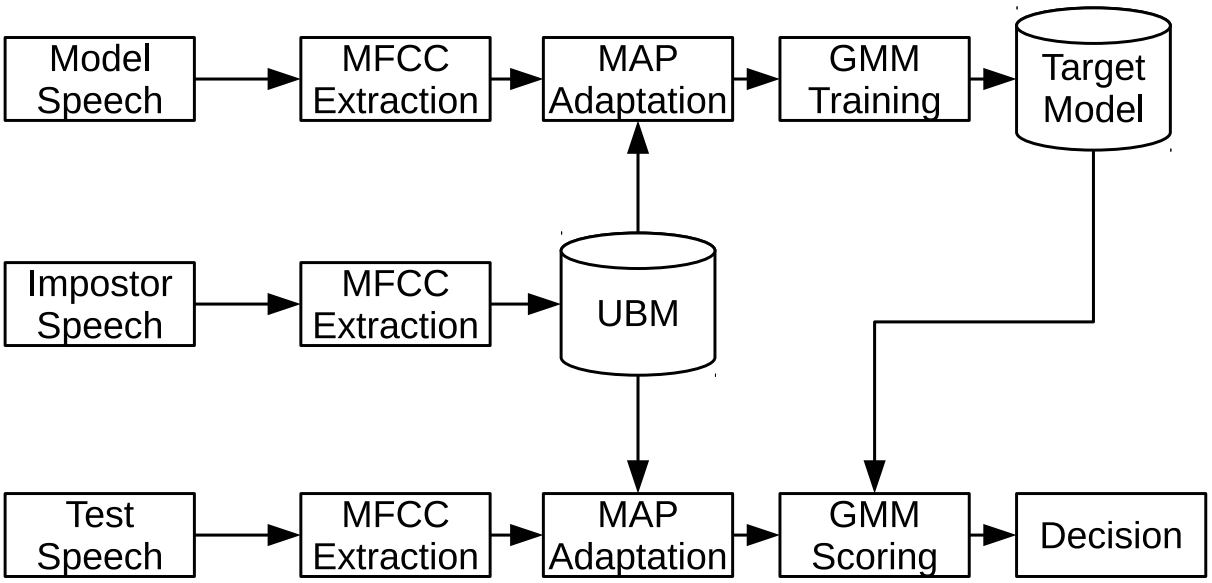


Figure 3.2: A block diagram of a GMM-based verification system.

3.3.2 SVM Classification

In SVM classification, the same feature vectors used GMM-UBM classification are used to train SVMs. The difference here is that instead of building a generative model to characterise a class, SVMs focus on the features that fall at the boundary between different classes. SVMs are binary classifiers, meaning that in order to classify languages, a single SVM can only find a hyperplane between a particular language and all other impostor languages. This is not sufficient for multi-class classification problems. There are two strategies usually employed to tackled this

problem. The first method is to construct multiple binary classifiers for all possible pairs of classes to identify one language from the other. The second method is to construct multiple one-against-all SVMs, where every SVM classifies one language against all other impostor languages. The chosen strategy is usually dependent on the data available. With a large volume of training data, a one-against-all SVM could have very long training times given a larger optimization problem. On the other hand a series of binary classifiers could more suitably deal with larger data sets. Whilst one-against-all SVMs seem more popular in speech classification literature, there is no general consensus as to which method is better. It is usually left as a choice for each particular system.

One important factor in performance of SVM classification is the choice of kernel to use for the acoustic features. It is very hard to classify say, a language, at the frame level. Therefore classifying frames individually with an SVM is not usually performed. Two popular kernel choices are the Fisher Kernel and the Generalized Linear Discriminant Sequence (GLDS) Kernel.

3.3.2.1 The Fisher Mapping Kernel

The Fisher mapping kernel is a popular kernel choice that combines generative models with discriminative training in SVMs [126, 127, 128]. The SVM input vectors are derived from the generative model itself. So in effect the class dependent GMMs are still derived by MAP-adapting a UBM for class-specific training data. The GMM plays the role of the generative model. If we assume a GMM class c parameterized by λ and an utterance sequence X, then the Fisher mapping kernel is based on the first derivative of the GMM likelihoods. It is obtained by Equation 3.10 [124].

$$f_{\text{fisher}}(X) : X \mapsto \nabla_{\lambda} \log P(X|c, \lambda) \tag{3.10}$$

Given this mapping function, the kernel score between two utterances is computed as in Equation 3.11 [124]. The term R is the covariance matrix of the data in the Fisher mapping space, and is determined by $R = E\left[f_{\text{fisher}}(X) f_{\text{fisher}}(Y)\right]$.

$$k_{\text{fisher}}(X, Y) = f_{\text{fisher}}(X) R^{-1} f_{\text{fisher}}(Y) \tag{3.11}$$

3.3.2.2 The Generalized Linear Discriminant Sequence Kernel

In sequence kernels, the basic approach is to compare two utterances by training a model on one utterance and then scoring the resulting model on another utterance. The GLDS is a linear kernel and given an utterance of frames X, then the mapping function of the GLDS kernel is expressed in Equation 3.12 [129, 124].

$$f_{\text{GLDS}}(X) : X \mapsto \frac{1}{N} \sum_{n=1}^{N} b(x_n) \tag{3.12}$$

The term $b(x_n)$ is the polynomial expansion of the speech frame. Given this mapping function, the kernel score between two utterances is computed as in Equation 3.13. The term R is the normalization matrix obtained by $R = M^t M$, and M is defined in Equation 3.14 [124]. The terms $f_{\text{GLDS}}(X^{C_1})$ and $f_{\text{GLDS}}(X^{Z_i})$ are the polynomial expansion of the class and impostor data sequences respectively. The terms N_C and N_I represent the number of class and impostor sequences

$$k_{\text{GLDS}}(X, Y) = f_{\text{GLDS}}(X) R^{-1} f_{\text{GLDS}}(Y) \tag{3.13}$$

$$\begin{bmatrix} f_{\text{GLDS}}(X^{C_1}) \\ f_{\text{GLDS}}(X^{C_2}) \\ \cdots \\ f_{\text{GLDS}}(X^{C_{N_C}}) \\ f_{\text{GLDS}}(X^{Z_1}) \\ f_{\text{GLDS}}(X^{Z_2}) \\ \cdots \\ f_{\text{GLDS}}(X^{Z_{N_I}}) \end{bmatrix} \tag{3.14}$$

A main characteristic of this kernel is that the average of all the projected vectors removes the context variability resulting from phonemic context, which results in loss of information. However this kernel was found to be useful for SID and LID problems.

3.3.3 GMM-SVM Classification

In GMM-SVM classification, the idea is to first model the sequence of acoustic vectors of an utterance as a GMM (adapted from the UBM), and then to define a kernel function that measures the similarity between different utterances (as GMMs), which satisfies the Mercer

conditions. The GMM is usually represented as a supervector, which is a high-dimensional vector made up of the concatenation of GMM component mean vectors. If M is the number of components of a GMM, and D is the dimensionality of the fronend feature vectors, then the supervector has a dimensionality of $M \times D$. The GMM-SVM classification scheme can be considered a direct extension of the GMM-UBM verification process as shown in Figure 3.3. As with SVM classification, it is important to make the right choice of kernel to leverage the use of GMM systems within an SVM framework that satisfies Mercer's conditions. The following sections show how the previously described Kullback-Leibler divergece used in the GMM-UBM framework is extended for SVM kernels.

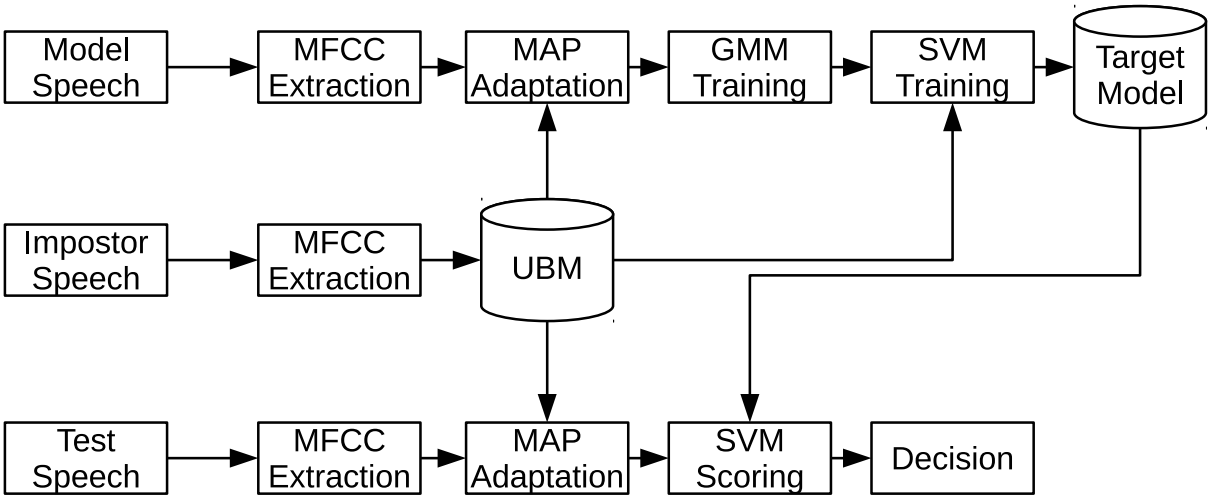


Figure 3.3: A block diagram of a GMM-SVM classification system.

3.3.3.1 Linear GMM-SVM Kernel

Given utterance based GMMs, there needs to be a way to score one utterance with an other to find out how similar they are. One way to measure the similiarity between GMMs is the Kullback-Leibler divergence. However, the Kullback-Leibler divergence does not satisfy the Mercer Conditions for SVM kernels because the kernel matrix of distances based on symmetric Kullback-Leibler divergences is not a positive definite matrix. However, there is an approximation to the Kullback-Leibler divergence to represent the distance between two GMMs which is suitable for a linear SVM kernel. This is given in Equation 3.15 [130, 131, 124].

$$\begin{aligned} k_{\text{linear}}(X, Y) &= \sum_{i=1}^{M} w_i \mu_i^X \Sigma_i^{-1} \mu_i^Y \\ &= \sum_{i=1}^{M} \left(\sqrt{w_i} \Sigma_i^{-1/2} \mu_i^X \right)^t \left(\sqrt{w_i} \Sigma_i^{-1/2} \mu_i^Y \right) \end{aligned} \tag{3.15}$$

The terms μ_i^X and μ_i^Y are the adapted means of GMM component i for utterances X and Y respectively. The terms w_i and Σ_i are the weights and variances of the UBM. The resulting supervectors from MAP-adapting a UBM to an utterance are used with Equation 3.15 to train an SVM per language in a one-against-all setup. During recognition, each test utterance is mapped to a supervector by MAP-adaptation of the same UBM. The resulting supervector is input to all the SVM language models, and a score is given. A positive output score indicates membership of the test utterance in a class, whilst negative scores indicate that the utterance belongs to the non-target classes.

3.3.3.2 Non-Linear GMM-SVM Kernel

During the period of development of the linear GMM-SVM kernel above, a new non-linear kernel was suggested based on the same symmetric Kullback-Leibler divergence. This kernel function is given in Equation 3.16 [132]. This non-linear kernel is equivalent to a Radial Basis Funtion (Gaussian) kernel applied in GMM supervector space.

$$k_{\text{nonlinear}}(X, Y) = \mathrm{e}^{-\sum_{i=1}^{M} w_i (\mu_i^X - \mu_i^Y)^t \Sigma_i^{-1} (\mu_i^X - \mu_i^Y)} \tag{3.16}$$

3.4 GID in Literature

In this section we give an overview of GID techniques and results from selected papers in literature. This overview is by no means exhaustive. The section goes over the breadth of different techniques covering state-of-the-art performance from many different contributions leading up to this thesis. The evaluations are performed in a very different fashion across the literature, so it is hard to draw a full comparison across these works. We try to group related studies together (though not necessarily chronologically) to help in this regard.

The problem of automatic gender identification in speech has been studied using various techniques. In [133] various features extracted from clean speech are considered (autocorrelation, linear prediction, cepstrum, and reflection). The features are applied specifically to represent vowels, voiced and unvoiced fricatives. The evaluation compared a number of distance measures, filter orders and recognition methods. The results shows that in general, all feature types are effective for GID, even when using a simple Euclidean distance measure, which gave the most robust results. The claim made is that gender is time invariant, independent of specific phonemes, and gender groups can be recognized quite easily without requiring speaker

compensation. This study achieved 100% GID accuracy over a database of 52 speakers. This early attempt at GID, though on a limited speaker corpus and controlled recording conditions, showed that GID is not a very hard task, and there is a certain flexibility as to which features can be used.

In another study that considers GID over short segments of sound [134], gives some surprising results from tests on the TIMIT corpus. Speech was represented by cepstral coefficients and first order delta coefficients. Classification was made on individual coefficients by a single Gaussian classifier. Since the TIMIT corpus is phonetically labelled, tests can be run for general speech, acoustic classes, or individual phonemes. The author expected that training on phonemes would give the best performance and superior to a classifier trained on general speech parameters. The results showed the contrary, where except for fricatives, training and testing by sound class was better than phoneme-based classification. Also, except for stop sounds, a system based on sound classes was better than training and testing on general speech. The GID classification rates ranged from 55% to 95%. Although the error rates were large in some cases, this work proves that GID can work, in some cases, on very short speech segments.

A study by [135] related to GID for different languages combines acoustic analysis and pitch. Genders are matched to HMMs from speech features using the Viterbi algorithm. The acoustic analysis extracts 12 cepstra, 12 transitional cepstra, energy and transitional energy. LDA was applied to the male and female models for speaker normalization. GID was performed by extracting frame features, and matching them to the male and female models (under the same LDA transformation). The model producing the most matches is used for the classification result. Testing on other languages than the one trained on gave low error rates of less than 5.2%, and an average of 2.0%. Another study revolving around language independence is [136], which describes a novel Gaussian Mixture Model (GMM) classifier based on a concatenation of pitch values with the corresponding RASTA-PLP feature vector. A small order GMM (4-8 components) is sufficient in their experiments. The results range from 98% for clean speech and goes down to 95% for the noisiest speech when degraded to a SNR of 0dB.

The work in [137] demonstrates 63 different GID systems based on the fusion of many knowledge sources using a linear classifier implemented as a simple perceptron rule-learner. The learner acts as a fusion system for multiple GMMs that model different speech features such as MFCCs, reflection coefficients, autocorrelation coefficients and log area ratios. The GID system is tested on different languages in the OGI speech corpus. The main result shows that training a classifier on a diverse set of languages (rather than one) improves GID classification.

This further suggests that the phonetic context, albeit at a general level, is useful in GID.

The work in [138] proposes an automatic gender identification algorithm based on building separate Hidden Markov Models (HMM) for the genders. This work makes the assumption that speakers in the training and testing sets have a closed vocabulary that they can use for utterances. With a closed vocabulary, it is possible to construct a HMM for each gender, based on the sequences of observations in the training set. In the test case, the utterance is then matched against both gender HMMs, and the HMM that gives the highest score is selected. Low error rates were reported in this experiment (2.4% for male speakers, and 6.1% for female speakers). The main problem with this approach however is that in normal conversational speech, the vocabulary is virtually unlimited, making gender identification systems built on closed vocabulary HMMs impractical. On the other hand, this work shows that knowing the context of a sound (via HMM states, in this case) has a strong impact on the performance of a gender identification system. In this approach, training was performed on a relatively low number of test samples, from a low number of speakers (8 males and 8 females).

In another study [139], the focus is solely on pitch and cepstral features namely LPCCs, MFCCs and PLPs. The purpose of this study was to measure the effect of which cepstral features are better across corpora with unmatched training and testing conditions. The results indicate that using voiced speech frames, and modelling higher order spectral detail (by using higher order cepstral coefficients) along with delta dynamics improve the GID robustness. Pitch is generally complementary to GID. However, results are better in noisy conditions if pitch is removed from the feature vector. The study in [140] attests to the complementarity of pitch for GID in clean conditions. The study looks at the effect of pitch, formants and combinations of both for GID. The authors recorded a ten Hindi digits database for fifty speakers. GID is tested separately for formants and pitch. Pitch was also tested by different extraction methods (autocorrelation, cepstrum and AMDF). The combination of formants with pitch information gave the best results. Moreover, a feature vector consisting of pitches from all methods together was also tested. For open-set testing, the autocorrelation pitch method performs best, whilst for closed-set tests, the combined feature vector gave best performance. The use of formants is also mentioned in the work by [141] which investigates the use of GID over a few sentences to perform model selection for speech recognition. The GID system employed is based on the location of the first two formants in the frequency domain. The analysis is based on a frequency bin that is common to both males and females, discovered from training data. Within that range, the position of the formants is then used as an indicator of gender during classification.

The introductory claim in[142] is that most of the current GID systems are not suitable for speech in audio-visual data, since a lot of assumptions are made about the quality of speech as well as the preprocessing required for silence removal, voiced speech detection and phoneme recognition. The authors use a general audio classifier that does not have any particular constraints on speech quality, segment lengths etc. Their system reaches an accuracy of 92%. They propose the use of the long term features for GID. The feature vectors are extracted over larger windows (1s), and are made up of the first order statistics of the signal's spectrum (a set of Mel Frequency Spectral Coefficients) gathered with a 10ms frame rate. The mean and variance of these are estimated over a long term window. The features are fed into a Multi Layer Perceptron Neural Network classifier. Moreover, the training data was split into different sets and many Neural Network experts are combined for classification. The test data contained recording from French and English radio stations, telephone speech, outdoor speech and studio speech. The accuracy of 92% is relatively high for the mixed conditions used for testing. However no comparison is made with a traditional short-term feature based GID classifier.

This section has reviewed a selection of published literature relating to gender classification. The comparisons between methods are hard to make. The corpora being utilised vary quite a lot in material and scope across these studies. Some of the conclusions made in some studies are local to a particular corpus and are not verified in a generic form. The work on GID seems to suggest that GID is not a hard problem. However, the different training and testing conditions require different feature sets, since there are issues of robustness when conditions change. Given the right feature sets, GID classification rates are quite high, with little problems related to speaker or language differences. Depending on the application area, there is also a division between using short-term, phoneme level feature vectors or longer term statistics. Again, these methods are not compared with each other, but rather confined to the particular corpora utilized in the respective papers. Our work on GID will try to show cross-corpora performance (on relatively well-known corpora) via our novel classification system, and assess what happens in these cases, and how to mitigate certain problems in a generically applicable framework.

3.5 AID in Literature

In this section we give an overview of AID approaches and results from selected papers in literature. There is an overlap in AID and LID, with LID being the more general field under which AID falls, and most AID techniques have come about from the development of LID.

This is not necessarily the correct approach to take, and there have, in fact, been some very interesting AID-specific techniques developed that do not feature in LID systems. However, AID has received more attention in the last few years than ever before, and has proven to be a very challenging problem for researchers. In our own work, speakers from an accent are chosen as a homogeneous population, having lived all their lives in a particular accent region. But the very concept of an accent requires us to think in terms of a continuous spatial representation of speakers within and across different accent regions. This is perhaps one of the main differences between AID and LID — the distinction is much more fine-grained in accents. Another thing to consider is that often in literature, reference is made to "dialect" to mean what we refer to as "accent" in this thesis, rather than our own definition of "dialect".

One of the earliest works in AID [143] describes an accent identification technique which aims to differentiate between various accents of Latin American Spanish speech. The problem is a two-class problem between Cuban and Peruvian accents, although the claim is that the technique could be extended easily to other accents. This work uses a PRLM system for identification. The phone recognizer is trained on English from the TIMIT corpus. The two-class problem here consisted of 143 Cuban and Peruvian speakers. The data was partitioned into three sets. Training is done on one of only two sets, and testing on the remaining two sets. The two sets used for training contained speech from speakers who were judged to be typical regional speech by each of the individual speakers. The third set was not used for training. Training and testing is performed on three-minute long utterances (with some speech removed since it belongs to the interviewer). The system reports an error rate of 16% on this two-class problem. Interestingly, the authors also introduce the creation of a new speech corpus to support future research — the "Miami" Latin American Spanish speech corpus. The problem, something that is still evident now, is the lack of corpora built with AID problems in mind, which have a good balance of number of speakers and accents.

Another early attempt at AID is the work in [144], which takes the acoustic approach, with text-dependency. The feature vector for this system comprised of mean cepstral and duration features per phoneme. Only a limited set of phonemes (primarily vowels) are considered. Data is collected from four regions. The classifier is a linear discriminant in this feature space to classify two broad regions of Northern US speakers from Southern US speakers, or a two-class problem between a pair of the four original regions. The authors report a 13% error rate in their experiments. The error rates for pair-wise classifications showed how the regions with least training data obtained the largest error rates of around 50%. A four-class problem, compensated

for gender, yielded an average error rate of 15%. When not compensated for gender, the error rate for the four-class problem goes up to 25%. Interestingly, the authors note in their conclusions that it seems hard to group speakers by the notion of geographical regions for speaker clustering. The more principled approach would probably be to replace this with abstract notions of similar speakers. The task would therefore not be one of classifying speakers into predetermined groups, but to cluster data into consistent groups.

The work in [145] takes on another interesting problem caused by accents, that of the influence of a foreign speaker's native language on his or her spoken English. In this work, a database is developed for words and phrases that are known to be sensitive to accent. The classifiers built are based on isolated words or phonemes, and the feature set used consists of MFCCs, energy, and the first order derivatives. The classification framework utilized is to input the extracted features into four separate Hidden Markov Models (left-to-right, with no state skips), one for neutral accents, and three others for Turkish, Chinese and German language accents respectively. The model giving the largest probability gives the classification. The accuracy of the system increased with longer utterance length, with isolated word strings of seven or eight words yielding a 93% AID rate for four different language accents. Training data came from 16 speakers, and testing was done on 12 speakers.

The work in [146] is interesting since it is concerned with two unsupervised approaches to AID. The techniques do not require a prior transcription for training. The comparison is between a low-level acoustic method based on mixture component usage, and a phonotactic system. In the first system, a set of Gaussian Mixture Models (256 components) are shared across a semi-continuous Hidden Markov Model. These components are used to model the complete speech space (all accents). At training time, for each speaker, the index of the most likely mixture-component for each frame is recorded, as well as the most likely state for the same frame. A new feature vector is created with the indices of the mixture components often used by the speaker in each state of each model. This is followed by speaker clustering by first creating a matrix of distances between all mixture components. For every speaker pair, a distance, per component, can be summed from this matrix. The speakers are clustered into as many groups as there are accents, and the centroids of the cluster represent the accent. This training phase clustering yielded one cluster with 29 American speakers and 13 British speakers, whilst the second cluster had 16 British speakers and no American speakers. At testing time, the same components index is collected, and then matched to the two clusters for nearest-neighbour classification. The second method is one based on diphone phonotactics and bigram language

modelling. The results go up to 96% accuracy for the mixture component usage method, and 100% for the phonotactic method.

More recent work [147] suggests the use of stochastic trajectory models for AID. This method is a text-independent system built on phone-based models. The idea is to capture the spectral evolution information as a cue for accent behaviour, in order to capture the coarticulation flow of an utterance. The speech utterance is converted into a stream of feature vectors, and then tokenized into a set of phone sequences. The sequence of tokens represents a trajectory. These trajectories are then mapped to different subspaces by PCA and LDA. Data is used for accents from speakers of British English, American English, Mandarin Chinese, French, Thai and Turkish. The results yields between 75% to 90% accuracy for pairwise classification tasks over isolated word concatenations, comparing neutral American English and a choice between Chinese, Thai or Turkish.

Another interesting take on AID, described in [148], is called Word-Based Dialect Classification (WDC) where the text-independent decision problem is converted into a word by word text-dependent decision problem. Every word is classified separately and the decisions are combined into a higher level classifier at the utterance level. A Hidden Markov Model is created for every word in a set of common words across all dialects. Transcripts for all words in all dialects are used to build a language model, which is used during testing to perform word recognition to pick only effective words from a test utterance. The language modelling is therefore task-dependent, albeit being dialect-independent. The word models are used to score the input words at test time as conditional probabilities. Finally the utterance classifier acts in this probability space as a majority vote over all words. Two-class classification is considered, and the classification error ranges from 1.6% to 3.4%. On a classification between eight dialects, classification error rates are in the range 20% to 26%.

One of the most recent interesting developments in AID is the ACCDIST metric by Huckvale [149]. The focus is to create a metric to quantify the accent distance between two speakers, rather than use data to learn the accent characteristics and then use standard metrics/classifiers over the model. The task was to create a metric that is very sensitive to, but at the same time uninfluenced by speaker and gender characteristics in speakers, even if operating from spectral envelope features. This technique was also tested on the ABI corpus, the corpus we use in this thesis for our experiments. In the first step, transcriptions of the utterances are generated from a trained Hidden Markov Model phoneme recognizer for Southern British English. Subsequent analysis was limited to vowel segments, which gave 145 vowel measurements per speaker in

the corpus. Formant locations were estimated by LP analysis. These formants were Z-score normalized per speaker. The spectral envelope for a duration of half a vowel was represented as the mean MFCC vector. In the next stage, an agglomerative bottom-up clustering method was used to combine speakers intro groups. Each speaker starts as their own sub-tree. At each iteration, two sub-trees are combined. The choice of which sub-trees are combined depends on the similarity between speakers and the linkage method. Single, complete and average linkage were tested. The distance metrics included correlation distance, Euclidean distance, weighted Euclidean distance and finally the ACCDIST metric. First, for each speaker N, a table of distances between all segment pairs i, j is calculated ($SS_{ij}^N = \text{dist}(s_i^N, s_j^N)$). Then these corresponding distance tables between speaker pairs are used to find the correlation between speakers, resulting in the ACCDIST metric (ACCDIST = $\text{corr}(SS^1, SS^2)$). The end result is a distance measure based on relative similarities rather than absolute properties. The speaker-specific segment pairs account for speaker variability, whilst the correlation accounts for speaker-wide differences, which in clusters, form accent specific distances. This metric gave very good cluster purity results when used with spectral envelope parameters and the complete or average linkage methods. Also, the work reports an AID recognition accuracy of 92.3% on the 14 accents of British English in the ABI-1 corpus.

We have already discussed the use of PRLM techniques and how they can be applied to AID. The work in [150] focuses on the idea that accents have finer-grained differences as opposed to languages, and suggests discriminative language models. The general framework is one based on converting speech to a sequence of tokens, and n-gram analysis is performed, and used with a sequence kernel SVM to model and predict classes. The paper then proposes two additions. The first is SVM feature selection, where an iterative wrapper on top of SVMs are applied to pick significant n-grams. For a set of features S, an SVM solution with model w is found. The features are ranked, and low ranking features are removed. The process is iterated multiple times. The resulting kernel is a sum of kernels up to the desired n. The SVM is trained and n-grams ranked according to their magnitude (sign of significance) of the entries in the SVM model w. The second addition revolves around the idea that higher order n-grams are more discriminative than lower order ones. Therefore, by first finding discriminative lower order n-grams, one can build very discriminative higher order n-grams with the already discriminative sub-sequences. Results are analysed for three dialects of neutral English, Mandarin and Arabic. Relative improvements of between 10% and 30% are recorded over standard PPRLM systems.

The work in [151] presents a study on five dialects of Arabic, using a phonotactic approach,

and resulting in an overall accuracy of 81.60% for 30s test utterances. The technique is a PPRLM system as described earlier in this chapter. The interesting aspect of this paper is the relatively larger test base compared to some of the earlier papers, and the five way classification problem (with a high accuracy) as opposed to smaller two or three way classification problems. The authors extend their work on this problem in [152]. This work focuses on the phones which are realized differently across dialects, using a kernel that computes phonetic similarity. The architecture is that of a phone-GMM-supervector based SVM kernel. Firstly, utterances are passed through a phone recognizer. For each phone, feature vectors for each frame within the phone instance are extracted. For each phone type, the features for the phone-type are used to train a GMM-UBM, across all dialects. With 34 phone types in the data, the result is 34 phone GMM-UBMs. Each phone GMM-UBM can be MAP-adapted to create a supervector for a dialect specific phone feature set. So each phone instance is represented as a supervector. An utterance is represented as a sequence of supervectors. SVMs are trained for each pair of dialects. The kernel used computes the similarity between pairs of utterances, as a sum of RBF kernel distances between all pairs of supervectors in a sequence. The order of supervectors in the sequence is unimportant. What is important is the the distance calculated over the supervectors of the same phone type. Classification follows the same supervector per phone extraction for an utterance. The final score is a sum over classifications of all supervectors in an utterance. Testing is compared for standard PRLM, standard GMM-UBM, the phonotactic approach in [151], and a discriminatively trained GMM-UBM system. The kernel method gives the best identification results, with an overall error rate of 4.9% for four Arabic dialects.

The recent work in [153] looks at the effect of analysing different frequency bands for SID and AID. The results are based on the ABI-1 corpus. The task here is not particularly to construct the best classifier for these tasks, but rather to evaluate processing the frontend using different configurations. Classification is based on GMM-UBM systems for both SID and AID. SID being the easier task, shows up to 100% accuracy when the entire bandwidth of the recorded signal is used (11.025kHz). At telephone band-pass filtered speech (0.23-3.4 kHz), SID accuracy goes down slightly to 97.54%. In contrast, the optimal AID performance of 60.34% is obtained when using band-pass filtered speech. The authors also test individual sub-bands, where the entire bandwidth is divided into 28 overlapping sub-bands. SID related information lies mostly in the regions of 0-0.77kHz (where primary vocal tract resonance information for vowels and nasal sounds appears) and 3.40-11.02 kHz (corresponding to high frequency sounds such as fricatives). On the other hand, for AID, related information lies in the region of 0.34-3.44kHz, a range where information about general voiced sounds is present. This information will always

be affected by the individual speaker and physiology, and it is therefore up to the classifier to try and attenuate these effects in favour of linguistic information related to accents.

Table 3.1: Performance in terms of Equal Error Rate (EER) and AID accuracy for the various systems in [4].

System	AID (ABI-1) 30s EER %	AID (ABI-1) 30s Acc. %	AID (ABI-1) SPA EER %	AID (ABI-1) SPA Acc. %
GMM-UBM (4096)	16.16	56.11	13.46	61.13
GMM-SVM (4096)	13.0	67.72	9.41	76.11
GMM-uni-gram	14.95	60.12	13.54	72.28
GMM-bi-gram	19.69	52.12	18.5	57.83
Acoustic-fused	12.33	73.6	8.3	77.32
Phonotactics	9.18	74.05	6.5	82.14
Acoustic-Phonotactics-fused	6.4	88.8	4.52	89.6
ACCDIST-Corr.	—	—	2.66	93.17
ACCDIST-SVM.	—	—	1.87	95.18

The most recent work that is relevant to this thesis also uses the ABI corpus [4] of 14 accents of British English. It is perhaps the most comprehensive and relevant set of results which we shall refer to in this thesis, together with the ACCDIST results. This work deals with LID, AID, and within-region dialects in Birmingham. The interesting part of the work for this thesis is that on AID. A number of methods are tested out, GMM-UBM, GMM-SVM, GMM-uni-gram, GMM-bi-gram, Acoustic-fused, Phonotactics, Acoustic-Phonotactics-fused, ACCDIST (original), and ACCDIST-SVM. We have already described the general ideas behind GMM-UBM and GMM-SVM systems. GMM-n-gram systems are the tokenized version of phonotactic SVM classifiers. Fusion is performed using Brummer's multi-class linear logistic regression toolkit. The phonotactic system is a PPRLM system based on fusing 16 different phonotactic systems (4 phone recognizers, with unigram, bigram, trigram, 4-gram SVM language models). The acoustic-phonotactics fused system is a combination of all systems together. The ACCDIST (original) system is included for reference in [4]. However, the authors suggest an extension, the ACCDIST-SVM method. In this system the speaker distance tables are averaged for a particular accent, and vectorized for SVM training and testing. A test speaker vectorized distance table is evaluated against all accent models. The correlation distance kernel is used for training and evaluating the SVM. A summary of the results obtained is shown in Table 3.1.

This section has reviewed a selection of published literature relating to accent classification. AID lacks a unified evaluation database and system, such as the Language Recognition Evaluation (LRE) for languages, and thus comparisons are hard to make across different methods. However there is a sense of continuity in techniques and experiments that one can get from the quasi-chronological overview we gave in this section. A lot of the work has focused

heavily on some form of transcription, or at the very least, a phone recognition system as part of the front-end. The systems that do not make use of such a front-end, in fact, perform very poorly on the AID problem.

3.6 Variability Compensation

We have only touched on some of the primary issues that significantly degrade the performance of AID systems. These are changes in channel, within-class speaker differences and noise. Techniques to compensate for these effects have evolved together with the mainstream techniques for SID, LID etc. We have described feature normalization techniques earlier on in this thesis. Other techniques focus on modifying the class model, or the class features prior to modelling. One of the earliest techniques to deal with inter-speaker variability within, say, a language for speech recognition is Vocal Tract Length Normalization (VTLN)[154], while compensating for noise can be done with techniques such as noise masking [155] and Parallel Model Combination (PMC) [156]. When it comes to classification tasks, it is, however, very important to maintain the class characteristics, rather than normalize the spectrum to conform to a a particular model for efficient speech recognition.

3.6.1 Score Normalization

Aside from feature normalization techniques, initial methods for classification problems revolved around score normalization [157, 158, 159]. Successful examples of these are Zero Normalization (or Z-Norm) and Test Normalization (or T-Norm). Both have been successful in SID as well as other classification problems. The idea of score normalization is that the raw verification score is normalized to a different set of background speakers called *cohorts*. Scores from a system are shifted towards a common range of values, so that a speaker-independent verification threshold can be estimated. Score normalization follows the form in Equation 3.17, where s' is the normalized score, s is the original score, and μ_I and σ_I are the mean and standard deviation of the cohort score distribution.

$$s' = \frac{s - \mu_I}{\sigma_I} \tag{3.17}$$

The cohort distribution statistical estimates vary depending on what kind of normalization scheme is used. In Z-Norm, μ_I and σ_I are computed offline during speaker enrolment, by matching non-target utterances against the target model, to obtain values of mean and standard

deviation. On the other hand in T-Norm, the parameters are computed on the fly during testing by matching the unknown speaker feature vectors against a set of cohort speaker models. Z-Norm and T-Norm can also be used in combination, usually producing better results [160]. A variation of Z-Norm and T-Norm is H-Norm (and HT-Norm) [161, 74, 162], which is a handset (or channel) oriented version of the two normalization methods. The process is the same except statistics are gathered on channel-dependent variants of the target or cohort models.

Score normalization, though effective to reduce verification error rates is highly dependent on the right cohort utterances being selected. If the acoustic and channel conditions are still too different from those of enrolment and testing utterances, then the effect can be detrimental. It is believed [163] that score normalization may be altogether removed if eigenchannel compensation of speaker (or class) models are well-optimized. On the other hand, for the deployment of verification systems based on Joint Factor Analysis (which we describe later on), score normalization is essential [164].

3.6.2 Model and Feature Mapping

Two techniques are proposed in [165] for SID. The first is called Speaker Model Synthesis. First, a channel independent root UBM is trained using data from many different channels. Channel-dependent data is then used to build GMM for a specific channel with MAP-adaptation. This means that there is a direct correspondence between the UBM components and the adapted channel GMM component. Transformations can be calculated between different channel GMMs by computing the mean shift, variance scale and weight scale that transforms one channel GMM into another channel GMM. During speaker enrolment, the most likely channel is assumed for a speaker, but transformations to all other channel GMMs are applied to the speaker GMM model. During testing, the most likely channel GMM is detected, and the speaker GMM for that particular channel is used. The second technique proposed in [165] performs feature mapping, which maps features from different channels into a common channel independent feature space. The appeal here is that features are aggregated into a single model, rather than requiring multiple models to be constructed. A channel-independent UBM is constructed using data from multiple channels. Channel dependent GMMs are constructed similarly to Speaker Model Synthesis. The feature space mapping between the original UBM and the channel GMMs can be calculated. Given enrolment data from a speaker, the most likely channel GMM is detected, and the mapping function calculated between this GMM and the UBM is used to map the features to create a channel-invariant speaker GMM. Results on a number of SID experiments

showed improvements over a baseline system. However the main drawback is the requirement of knowing the channel type for training data.

3.6.3 Inter-Session Compensation

One of the most recent techniques suggested to work very well for LID [166], and also the method of choice for the AID work in [4, 153, 167] is that of Inter-Session Compensation. Within the context of LID, we can consider many factors causing variability, such as different speakers within a language, different channels, noise, utterance length etc. The idea of Inter-Session Compensation (ISC) is that the variability present in the high dimensional supervector space can be represented by a small number of parameters in a much lower dimensional subspace. These parameters are called the channel factors [168]. In [166], ISC is applied to the feature domain, as opposed to the original suggestion of model domain adaptation in [168, 169]. Basing the work on the GMM-UBM framework, model domain compensation is achieved by shifting the mean supervector of the UBM, together with the language-dependent GMM supervectors towards an inter-session variability direction which is estimated from the test utterance, as shown in Equation 3.18 [167].

$$\bar{\boldsymbol{\mu}}_{\text{sv}} = \boldsymbol{\mu}_{\text{sv}} + Ux \tag{3.18}$$

The terms $\bar{\mu}_{\text{sv}}$ and μ_{sv} are the shifted and original supervectors respectively. The term x is an R-dimensional vector comprising the channel factors for the test utterance whilst U is a low rank matrix projecting the channel factors x from the low-dimensional channel factor subspace to the high-dimensional supervector domain. U is referred to as the *Eigen-channel subspace matrix*.

The directions (eigen-vectors) where the supervector are mostly affected by inter-session variation, are defined by a $CF \times R$ matrix U where C, F and R are the number of GMM components, the feature dimensionality, and the chosen number of eigen-vectors. R is usually taken to be $\geq$50. Therefore, U is given by a set of R eigen-vectors of an average within-class covariance matrix. Each class is represented by supervectors estimated from utterances of that class. In the case of LID, for each language l, and each utterance of the language $\{j = 1, \ldots, J_l\}$, the UBM is adapted by the utterance to obtain supervectors for the utterances, denoted by sl_j. The average supervector for a language is then given by Equation 3.19 [167].

$$\bar{s}_l = \frac{1}{J_l} \sum_{j=1}^{J_l} sl_j \tag{3.19}$$

The average supervector $\bar{s}_l$ is subtracted from each of the supervectors of the language sl_j. The resulting vectors are formed into columns of an $CF \times J$ matrix S, where J is the number of all utterances from all languages. The model assumption is that as $\hat{s}_l$ is subtracted from sl_j, the resulting supervector is due to inter-session variability. The columns of the matrix U are given by the R eigen-vectors of the covariance matrix SS^T ($CF \times CF$), corresponding to the R largest eigenvalues. However, for large component GMMs, using PCA to compute the eigen-vectors can be infeasible.

Once the eigen-channels are estimated, a GMM for every language can be adapted to the channel of the test utterance by shifting the supervector in the directions dictated by the eigen-channels, to better fit the test utterance data. If s is the supervector representing the model to be updated, and $P(O|s + Ux)$ is the likelihood of the test utterance $O = o_1, o_2, \ldots, o_T$, given the adapted supervector, then it can be shown [169] that the value of x is given by Equation 3.20 [167].

$$x = A^{-1} \sum_{c=1}^{C} U_c^T \sum_{t=1}^{T} \gamma_t(c) \frac{\boldsymbol{o}_t - \boldsymbol{\mu}_c}{\sigma_c} \tag{3.20}$$

The term U_c represents the $F \times R$ section of the entire U matrix for the c^{th} GMM component. The term $\gamma_n(c)$ is the probability of mixture component c for the frame n, whilst μ_c and σ_c are the c^{th} mixture component mean and standard deviation. The term A is defined by Equation 3.21 [167].

$$A = I + \sum_{c=1}^{C} U_c^{'} U_c \sum_{t=1}^{T} \gamma_t(c) \tag{3.21}$$

The above ISC method is a very effective method to reduce the effects of channel variability. It is applied during recognition given an arbitrary test utterance. Of course, different GMM configurations require different estimates of U. However, the technique can be applied at a feature level, prior to building class GMMs for training channel compensated class models. The compensated feature frame is given by Equation 3.22 [167]. This way, frames for training data are also compensated for channel differences, as well as those at testing time. This allows the construction of GMM-SVM systems to be used with ISC.

$$\hat{\boldsymbol{o}}_t = \boldsymbol{o}_t - \sum_{c=1}^{C} \gamma_t(c) U_c \boldsymbol{x} \tag{3.22}$$

3.6.4 Joint Factor Analysis

In the ISC technique we focused on how supervectors, or features, can be compensated for session variability, and this has been useful in SID and LID applications, as well as newer AID applications. In this section we discuss a more complete setup, based on generative modelling to reduce various forms of variability. This technique is based on GMM and factor analysis, and called *Joint Factor Analysis* (JFA) [170, 124].

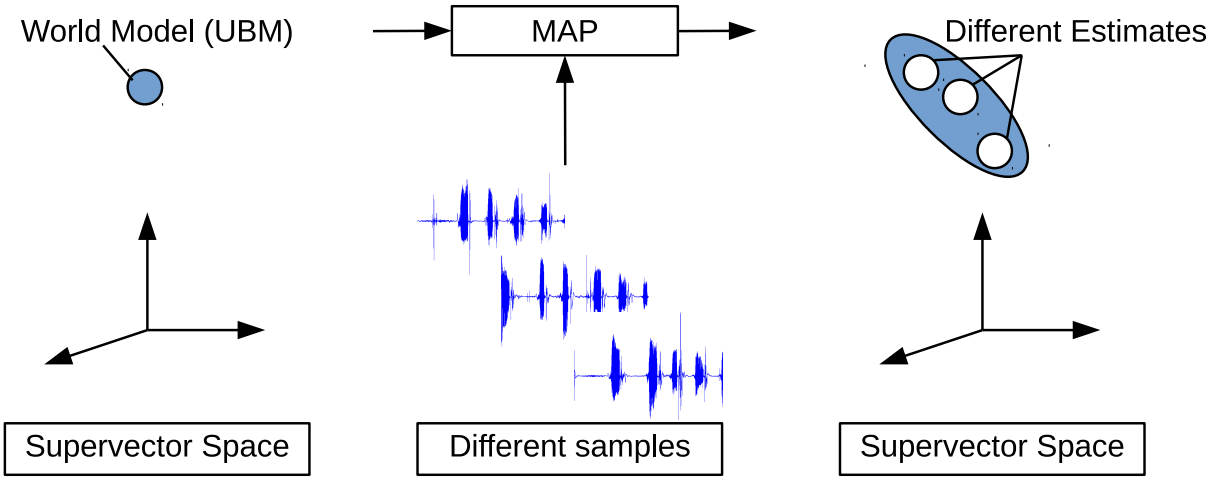


Figure 3.4: Traditional MAP adaptation producing rough perturbations of the actual class.

In traditional MAP adaptation, as shown in Figure 3.4, the mean vectors of a UBM are adapted to speaker specific data to build a speaker model, whilst variance and weights are usually left unadapted and shared across all speakers. So, the speaker can be represented as the concatenation of mean vectors, which we call the supervector. A speaker may have different training utterances, and the resulting supervectors will not be the same, and are therefore not representing the actual speaker, especially when the recordings come from different channels. Since training and testing may be altogether performed on different channels, we have seen that it is important for channel variability to be compensated in order to have consistent scoring of speaker models. To do this, the channel variability needs to be modelled explicitly, rather than incorporated into a monolithic model with the speaker information.

JFA considers the variability of a Gaussian supervector as a linear combination of the speaker and channel components. For a training utterance, the resulting GMM supervector M can therefore be decomposed into two statistically independent components as in Equation 3.23, where s and c refer to the speaker and channel supervectors, respectively.

There is no prior justification to consider channel and noise factors as existing solely in separate subspaces to the speaker voice factors. This is an educated assumption of the JFA

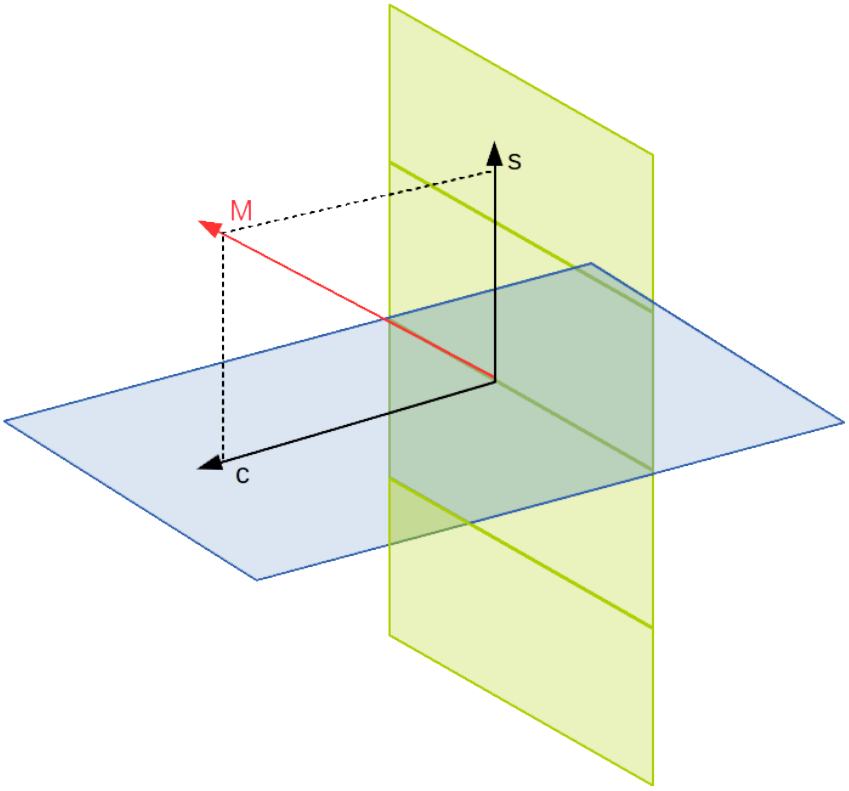


Figure 3.5: Decomposition of the supervector M into speaker s and channel c components by factor analysis.

model. For any given speaker, the speaker factors are assumed to be constant for all recordings of that speaker, but the channel factors may or may not change for each and every recording. Factor analysis is used (instead of PCA on GMM supervectors), to find the subspace components, since the random vector M is not observable from the data (or hidden), and there is no analytical way of estimating s and c.

$$M = s + c \tag{3.23}$$

The assumption in Equation 3.23 is that the speaker and channel effects lie in different subspaces of the supervector space. This is what makes factor analysis decomposition practicable. The concept is shown in Figure 3.5, where the speaker component s lies in a speaker subspace of two dimensions, and the channel component c lies in a channel subspace of two dimensions whilst the supervector M lies in the supervector space of three dimensions.

The details of JFA can be a bit complex to understand. Before describing the JFA paradigm, it will be useful to define a number of terms and ideas. Most of the operations to create a JFA system are defined in terms of *sufficient statistics*. These provide complete information of an arbitrary utterance needed to compute an estimate of GMM parameters [171]. If we assume that the set of utterances of a speaker s is given as a sequence of T feature vectors of dimensionality

F, represented as a $F \times T$ matrix of observations $O = [\boldsymbol{o}_1, \ldots, \boldsymbol{o}_T]$, then the alignments of each frame to each GMM component c are defined by the sufficient statistics in Equations 3.24- 3.26. These are referred to as the zero, first, and second order statistics respectively.

$$N_c(s) = \sum_{t=1}^{T} \gamma_t(c) \tag{3.24}$$

$$F_c(s) = \sum_{t=1}^{T} \gamma_t(c)\boldsymbol{o}_t \tag{3.25}$$

$$S_c(s) = \sum_{t=1}^{T} \gamma_t(c)\boldsymbol{o}_t\boldsymbol{o}_t' \tag{3.26}$$

These component based statistics can be expanded in supervector form as in Equations 3.27-3.29.

$$NN(s) = \begin{bmatrix} N_1(s)I & 0 & \cdots & 0 \\ 0 & N_2(s)I & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & N_C(s)I \end{bmatrix} \tag{3.27}$$

$$FF(s) = \begin{bmatrix} F_1(s) \\ \vdots \\ F_C(s) \end{bmatrix} \tag{3.28}$$

$$SS(s) = \begin{bmatrix} S_1(s) & 0 & \cdots & 0 \\ 0 & S_2(s) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & S_C(s) \end{bmatrix} \tag{3.29}$$

MAP adaptation can now be redefined in terms of sufficient statistics in a closed-form solution as shown in Equation 3.30. The term $\mu_{\mathrm{ML}}^{(c)}$ is the maximum likelihood estimate of the mean (weights and covariance are fixed to the UBM values), and τ is the relevance factor that controls the degree of interpolation between the UBM and the adapted GMM (e.g. it takes τ frames to move the parameter values half way between the UBM and the ML estimate). We shall redefine this kind of MAP adaptation as relevance-MAP adaptation [124].

$$\mu_{\mathrm{MAP}}^{(c)} = \beta^{(c)}\mu_{\mathrm{ML}}^{(c)} + (1 - \beta^{(c)})\mu_{\mathrm{UBM}}^{(c)} \tag{3.30}$$

where

$$\beta^{(c)} = \frac{N_c(s)}{N_c(s) + \tau} \tag{3.31}$$

and

$$\mu_{\text{ML}}^{(c)} = \frac{1}{N_c(s)} F_c(s) \tag{3.32}$$

3.6.4.1 ML-Trained MAP Adaptation

For this relevance-MAP adaptation technique, the prior distribution of a GMM speaker supervector s is normally distributed with mean vector $\text{E}[s] = \mu_{\text{UBM}} = m$, and a covariance diagonal matrix $\text{Cov}(s, s) = \frac{1}{\tau}\Sigma$. In relevance-MAP, the value of τ is found empirically. The work in [172, 173] proposed a way to find a ML-based estimation of the a priori variance of the speaker population with a training corpus. In this new model, the supervector s for an arbitrary speaker is written in the form of hidden variables as in Equation 3.33 [124].

$$s = m + Dz \tag{3.33}$$

The term m is still the speaker and channel-independent UBM mean supervector of dimension CF. The vector z is a hidden vector of dimension CF, which has a standard normal distribution, and D is a diagonal matrix with a dimensionality of $CF \times CF$. To obtain the posterior distribution that defines the supervector s, we require to know the a priori probability of the supervector s. With a prior normal distribution of this supervector (like in relevance-MAP), we can define the following two derivations in Equation 3.35 and Equation 3.40 [124].

$$\text{E}[s] = \text{E}[m + Dz] \tag{3.34}$$

$$= m + D\text{E}[z] \tag{3.35}$$

$$\text{Cov}(s, s) = \text{E}[(s - \text{E}[s])(s - \text{E}[s])'] \tag{3.36}$$

$$= E[(Dz - D\text{E}[z])(z'D' - \text{E}[z]'D')] \tag{3.37}$$

$$= E[Dzz'D' - Dz\text{E}[z]'D' - D\text{E}[z]z'D' + D\text{E}[z]\text{E}[z]'D'] \tag{3.38}$$

$$= D\text{E}[(z - \text{E}[z])(z - \text{E}[z])']D' \tag{3.39}$$

$$= D\text{Cov}(z, z)D' \tag{3.40}$$

Since the prior distribution of the hidden variable z is known to be a standard normal distribution, the mean vector and covariance matrix of the a priori distribution of supervector s are simplified as in Equation 3.41 and Equation 3.42 [124].

$$\text{Prior expectation of } s = m \tag{3.41}$$

$$\text{Prior covariance matrix of } s = DD' \tag{3.42}$$

The matrix D is derived from the a priori distribution of speaker supervectors, estimated iteratively from a training corpus of recordings with multiple recordings per speaker. Given the model parameters m and D and speaker training samples, the posterior distribution to calculate s is based on the posterior probability of the hidden variable associated with the particular speaker. The posterior distribution of the supervector s is therefore modelled in the same way as for Equation 3.35 and Equation 3.40, this time based on the posterior (rather than the prior) of the latent variable z with a mean vector E[z] and covariance matrix Cov(z, z). The term E[s] of the speaker posterior probability is the new GMM supervector estimated without relying on the relevance factor. This type of modelling, unlike relevance-MAP takes into account the uncertainty of the estimation of the speaker GMM, which is modelled explicitly with the covariance matrix Cov(s, s). The more training data is available, the less the influence of Cov(s, s). Provided D is well-conditioned with a small determinant, this form of MAP adaptation is equivalent to Maximum Likelihood training of the speakers when sufficient data is available for adaptation.

3.6.4.2 Eigenvoices MAP Adaptation

Another form of MAP-adaptation was found to be much more effective with short training samples than relevance-MAP [174]. Recalling the fact that for a speaker GMM, we can summarize the speaker model by a supervector of CF dimensions, the idea behind eigenvoice modelling is that PCA can be used to constrain this large supervector space (which requires a large amount of training data for proper adaptation from the UBM) to a much smaller subspace, with little loss of accuracy. The much smaller subspace, in turn, requires much less training data for proper adaptation. If we consider M_O and B to be the mean and covariance matrix of all the supervectors for a given speaker population, then the assumption is that most of the eigenvalues of B are near zero, and are unimportant for speaker modelling. The eigenvectors of B that correspond to nonzero eigenvalues define the *eigenvoices* of the population. The problem is

therefore to impose a constraint that a speaker supervector lies in the eigenspace.

The mechanism of eigenvoice adaptation is similar to the ML-based MAP adaptation. The assumption of the model is that the speaker space can be represented by a low rank rectangular matrix V of dimension $CF \times R$, where $R \ll CF$. The supervector s for a speaker is therefore defined as in Equation 3.43 [124]. Again, m is the UBM mean supervector.

$$s = m + Vy \tag{3.43}$$

The term y is the hidden vector of dimension R with a standard normal prior distribution. Similarly to the case of Equation 3.33, the expectation and covariance matrices of the prior distribution of s is defined by Equation 3.44 and Equation 3.45 [124].

$$\text{Prior expectation of } s = m \tag{3.44}$$

$$\text{Prior covariance matrix of } s = VV' \tag{3.45}$$

Again, the prior distribution of the supervector s is used to estimate the posterior distribution. Also, the eigenvoice adaptation models the mean vector $\mathrm{E}[s]$ and covariance matrix $\mathrm{Cov}(s, s)$ as in Equation 3.46 and Equation 3.47, respectively [124].

$$\mathrm{E}[s] = m + V\mathrm{E}[y] \tag{3.46}$$

$$\mathrm{Cov}(s, s) = V\mathrm{Cov}(y, y)V' \tag{3.47}$$

One other strong point of eigenvoice adaptation is that it models correlations between GMM components (bound by the factors in the subspace), whereas MAP adaptation does not (acting in the full supervector space), and therefore non-observed Gaussians are also adapted. A problem with this method however is that the rank of R is less or equal to the number of speakers available in the training corpus. There must be a significant amount of speaker diversity in the training corpus for a proper estimation.

With both traditional MAP and eigenvoices having advantages and disadvantages, they can be combined linearly to complement each other. The supervector for a speaker is therefore now defined as in Equation 3.48 [124].

$$s = m + Vy + Dz \tag{3.48}$$

3.6.4.3 Eigenchannel Model

In the previous section in ISC, we have seen that it is important to model session variability. So far in the JFA model, we have only spoken about the speaker model. JFA can be easily extended to support session variability explicitly as well as shown in [164]. Similar to the eigenvoice model of speaker space, we want to model the channel space, via a channel supervector c written in latent variable form as in Equation 3.49 [124].

$$c = Ux \tag{3.49}$$

As for Vy in eigenvoice modelling, U is a low-rank rectangular matrix of dimension $CF \times R_c$, where $R_c \ll CF$. The columns represent the eigenvectors of the channel covariance matrix, and defines the channel space. The hidden variable x has a standard prior normal distribution, and the components define the channel factors. Following the naming convention of eigenvoice adaptation, this channel variability model is called eigenchannel adaptation, and follows the same exact training procedure as eigenvoice training. By combining the channel variability with the rest of our speaker supervector definition, the result is shown in Equation 3.50 [124], which is the full specification of the JFA paradigm.

$$s = m + Vy + Ux + Dz \tag{3.50}$$

3.6.4.4 JFA Training Procedure

The matrices U, V and D are called the *hyperparameters* of the JFA model, and are estimated beforehand on large datasets. In [170, 164] is suggested to calculate V, then U, then D. The full JFA decomposition is shown in Equation 3.50. Training the JFA hyperparameters is done by:

1. Training the eigenvoice matrix V, assuming that U and D are zero.
2. Training the eigenchannel matrix U given the estimate of V, and assuming D is zero
3. Training the residual matrix D, given the estimates of V and U.

The next sections will show a step by step calculation of the JFA model via the sufficient statistics defined earlier, and are a reproduction of the procedure as outlined in [175].

3.6.4.5 Training the V matrix

1. $l_v(s) = I + V' * \Sigma^{-1} NN(s) * V$ (V is randomly initialised)
2. $y(s)$ $\mathrm{N}(l_v^{-1}(s) * V' * \Sigma^{-1} * FF(s), l_v^{-1}(s))$
3. $\bar{y}(s) = \mathrm{E}[y(s)] = l_v^{-1}(s) * V' * \Sigma^{-1} * FF(s)$
4. $N_c = \sum_s N_c(s)$
5. $A_c = \sum_s N_c l_v^{-1}(s)$
6. $\mathrm{C} = \sum_s FF(s) * (l_v^{-1}(s) * V' * \Sigma^{-1} * FF(s))'$
7. $NN = \sum_s NN(s)$
8. $V = \begin{bmatrix} V_1 \\ \vdots \\ V_C \end{bmatrix} = \begin{bmatrix} A_1^{-1} * \mathrm{C}_1 \\ \vdots \\ A_C^{-1} * \mathrm{C}_C \end{bmatrix}$ where $\mathrm{C} = \begin{bmatrix} \mathrm{C}_1 \\ \vdots \\ \mathrm{C}_C \end{bmatrix}$
9. update covariance $\Sigma = NN^{-1}\left(\left(\sum_s SS(s)\right) - diag((C) * V')\right)$
10. Run around 20 iterations of steps 1-9 with new estimates of V and Σ

3.6.4.6 Training the U matrix

1. Compute estimate of speaker factor y for each speaker
2. Compute 0^{th} sufficient statistic for each utterance (utt) of each speaker (s) in training data
3. $N_c(\mathrm{conv}, s) = \sum_{t \in \mathrm{conv}, s} \gamma_t(c)$
4. Compute 1^{st} sufficient statistic for each utterance (utt) of each speaker (s) in training data
5. $F_c(\mathrm{utt}, s) = \sum_{t \in \mathrm{utt}, s} \gamma_t(c) o_t$
6. For each speaker (s) compute the speaker shift using matrix V and speaker factors y
7. $\mathrm{spkrshift}(s) = m + V * y(s)$
8. For each utterance of each speaker in training, subtract Gaussian posterior-weighted speaker shift from first order sufficient statistics

9. $\bar{F}_c(\text{utt}, s) = F_c(\text{utt}, s) - \text{spkrshift}(s) * N_c(\text{utt}, s)$

10. $NN(\text{utt}, s) = \begin{bmatrix} N_1(\text{utt}, s) * I & & \\ & \ddots & \\ & & N_C(\text{utt}, s) * I \end{bmatrix}$

11. $FF(\text{utt}, s) = \begin{bmatrix} \bar{F}_1(\text{utt}, s) \\ \vdots \\ \bar{F}_C(\text{utt}, s) \end{bmatrix}$

12. Use $NN(\text{utt}, s)$ and $FF(\text{utt}, s)$ to train U and x in the same way we used $NN(s)$ and $FF(s)$ to train V and y (again, around 20 iterations)

3.6.4.7 Training the D matrix

1. For each speaker (s) compute the speaker shift using matrix V and speaker factors y
2. $\text{spkrshift}(s) = m + V * y(s)$
3. For each utterance (utt) of speaker (s), compute the channel shift using matrix U and channel factors x
4. $\text{chanshift}(\text{utt}, s) = U * x(\text{utt}, s)$
5. For each speaker in training, subtract Gaussian posterior-weighted speaker shift *and* channel shifts from first order sufficient statistics
6. $\bar{F}_c(\text{utt}, s) = F_c(\text{utt}, s) - \text{spkrshift}(s) * N_c(\text{utt}, s) - \sum_{\text{conv} \in s} \text{chanshift}(\text{conv}, s) * N_c(\text{conv}, s)$
7. $NN(s) = \begin{bmatrix} N_1(s) * I & & \\ & \ddots & \\ & & N_C(s) * I \end{bmatrix}$
8. $FF(s) = \begin{bmatrix} \bar{F}_1(s) \\ \vdots \\ \bar{F}_C(s) \end{bmatrix}$
9. $l_D(s) = I + D^2 * \Sigma^{-1} NN(s)$ (D is randomly initialised)
10. $z(s)$ $\text{N}(l_D^{-1}(s) * D * \Sigma^{-1} * FF(s), l_D^{-1}(s))$
11. $\bar{z}(s) = \text{E}[z(s)] = l_D^{-1}(s) * D * \Sigma^{-1} * FF(s)$

12. $N_c = \sum_s N_c(s)$
13. $a = \sum_s NN(s) * l_D^{-1}(s)$
14. $b = \sum_s FF(s) * (l_D^{-1}(s) * D * \Sigma^{-1} * FF(s))'$
15. $NN = \sum_s NN(s)$
16. $D = \begin{bmatrix} D_1 \\ \vdots \\ D_C \end{bmatrix} = \begin{bmatrix} a_1^{-1} * b_1 \\ \vdots \\ a_C^{-1} * b_C \end{bmatrix}$ where $b = \begin{bmatrix} \mathrm{b}_1 \\ \vdots \\ \mathrm{b}_C \end{bmatrix}$
17. Use about 10-20 iterations of steps 10-18 with new estimates of D

3.6.5 The i-Vector Model

The work in this thesis does not make use of the JFA model as presented in the last section. However the detailed overview of this model is given because it was extended in [176, 177] into another model, which we make use of for our AID research. We have seen how the JFA model defines a number of different subspaces to model speaker and channel variability. The approach proposed in [176, 177] is that of defining a single subspace, which contains all modes of variability. This subspace is called the *total variability* space. It is defined by a total variability matrix which contains the eigenvectors corresponding to the largest eigenvalues of the total variability covariance matrix. Contrary to JFA, there is no distinction between sources of variability from speaker effects or channel effects that influence the GMM supervector space. For a given utterance, the speaker- and channel-dependent GMM supervector M is defined in Equation 3.51 [124].

$$M = m + Tw \tag{3.51}$$

As in the JFA model, m represents the UBM mean supervector. The matrix T is the factor loading matrix of low rank, and w is a random vector having a standard normal distribution $\mathrm{N}(0, I)$. Following the style of definitions for the JFA model, the vector M is assumed to be normally distributed with a mean m and a covariance TT'. The process for training T is exactly the same as the eigenvoice matrix V training in JFA, with one difference. During eigenvoice training, the number of recordings from a particular speaker (or class) are considered to be as such i.e. from the same speaker/class. In total variability training, each utterance is considered to

come from a different speaker/class. Training the total variability space is now straightforward to understand. The same sufficient statistics calculated for JFA are gathered (separating utterances as being all from different speakers), and the matrices $NN(s)$,$SS(s)$ and $F(s)$ are defined as in JFA. The following section shows a step by step calculation of the total variability model via the sufficient statistics defined earlier, and are a reproduction of the procedure as outlined in [175].

3.6.5.1 Training the T matrix

1. $l_T(s) = I + T' * \Sigma^{-1} NN(s) * T$ (T is randomly initialised)
2. $w(s) \sim \mathrm{N}(l_T^{-1}(s) * T' * \Sigma^{-1} * FF(s), l_T^{-1}(s))$
3. $\bar{w}(s) = \mathrm{E}[w(s)] = l_T^{-1}(s) * T' * \Sigma^{-1} * FF(s)$
4. $N_c = \sum_s N_c(s)$
5. $A_c = \sum_s N_c l_T^{-1}(s)$
6. $\mathrm{C} = \sum_s FF(s) * (l_T^{-1}(s) * T' * \Sigma^{-1} * FF(s))'$
7. $T = \begin{bmatrix} T_1 \\ \vdots \\ T_C \end{bmatrix} = \begin{bmatrix} A_1^{-1} * \mathrm{C}_1 \\ \vdots \\ A_C^{-1} * \mathrm{C}_C \end{bmatrix}$ where $\mathrm{C} = \begin{bmatrix} \mathrm{C}_1 \\ \vdots \\ \mathrm{C}_C \end{bmatrix}$
8. Use abot 10-20 iterations of steps 1-7 with new estimates of T.

The motivation for containing all the variability within one subspace is that the experimentation performed in [124] showed that the assumption that channel factors of the JFA system normally model only channel effects was not entirely correct, and that some speaker information was also contained in this subspace. The vector w is a random variable, and the posterior mean of it estimated from an utterance is termed an i-Vector. Unlike the JFA supervector, it is not compensated for channel effects. In the i-Vector paradigm, channel compensation is carried out in the total factor space rather than in the GMM supervector space.

The advantage of applying channel compensation in the total factor space is the low dimension of these vectors, resulting in much less computation than for JFA modelling. The work in [177] proposes three ways of channel compensation: LDA, Within-Class Covariance Normalization (WCCN) [178], and Nuisance Attribute Projection (NAP) [179]. For the purposes of speaker verification, it was found that LDA followed by WCCN gave the best performance [177,

180]. As a final note, the i-Vector paradigm has since become the state-of-the-art model for many tasks such as SID and LID [124, 177, 180, 181, 182, 183]. In this thesis, we shall also apply and study the behaviour of the i-Vector paradigm for the AID problem. We will show that, with some additions, it can achieve state-of-the-art performance for this problem as well.

3.7 Prosody and Supra-Segments

In this section, we discuss some interesting work that relates to the analysis of rhythmic and prosodic information in speech in accents and languages. Some of the work we discuss here have not been used explicitly in LID or AID systems. However, features extracted from this work could add additional information to the feature extraction stage of these systems. In fact, we take inspiration from this work, and the thesis will report on results obtained for some of the rhythmic and prosodic information extracted from speech.

As mentioned earlier, we aim to build an AID system using approaches that do not rely on phonological language model knowledge of the accents, whether the transcription of phonemes is acquired in supervised or unsupervised form. Some research has shown that knowledge of the language for which accents are being analysed is not crucial in identifying accents. In [184], experimental evaluation showed that American human listeners performed only slightly better than non-native English speakers at classifying three British accents, as opposed to British listeners who outperformed all others by a wide margin. American listeners have linguistic knowledge of the English language, but this does not help much in AID tasks compared to speakers who are not speakers of English. Rather, British speakers, more accustomed to the accents (rather than just the language), performed better in these tests. Whilst linguistic knowledge has an effect on accent perception, the AID task seems to be partly acoustic problem, and perhaps AID does not depend on a prior tokenization of the speech signal. Another interesting set of findings is documented in [185] where the authors perform experiments in which speech utterances from different languages are either preserved or degraded in a number of ways. The findings support the idea that syllabic rhythm is necessary (and sufficient) for French adult subjects to discriminate English from Japanese.

Although some languages can be roughly differentiated, even by tamarin monkeys, the work in [186] demonstrates that greater neural activity is found for native-language speakers for perceptual identification, consistent with the hypothesis that native-language speakers use auditory phonetic representations more extensively than second-language speakers. This may

aid the identification of the subtle differences in accents as opposed to languages, corroborating the findings in [184].

The study in [187] shows how only a few studies in ASR have departed from the traditional Hidden Markov Model-based architecture with short-term feature vectors. The suggestion is that knowledge of the speech production process should be used to a larger degree, and that there has been limited research in the field with some interesting results. The authors review a number of works that attempt to map the traditional acoustic features to multiple streams of acoustic-only, acoustic-articulatory, articulatory-only features. The task of mapping acoustic features to articulatory features is however, not trivial. We can record acoustic information easily, but it is very impractical to record articulatory information. Therefore, researchers have focused on suggesting techniques for one-to-many acoustic-to-articulatory mappings, such as in [188, 189, 190].

There has been work investigation the classification of languages from rhythmic and supra-segmental features from speech such as [191, 192], extracted automatically, whilst others are done based on hand-labelled data [193, 194]. The common problem is always that of segmenting speech into correct supra-segmental units. There seems to be general agreement about correlates between the speech signal and linguistic rhythm [195], but reaching a consistent vector representation is difficult. The main ideas have focused around segmenting speech into short segments (bursts) that contain transient parts of speech such as coarticulatory behaviour. Following the extraction of these segments, combinations that contain transitions over time between consonants (C), vowels (V) and back to consonants (C) can be summarised by statistical representations. For this, vowel detection is necessary. The combinations of C-V-C transitions are normally termed *pseudo-syllables*. The automatic extraction of these transitions gives no guarantee of actual syllables being present. There have been reported relatively good results on LID with these features, based on traditional modelling and classification systems. However, taking into account speakers, results are inferior, showing that even at the supra-segmental level of analysis, the features are correlated with speaker behaviour (such as tempo). This kind of analysis is also mentioned (lightly) in relation to the AID problem, though in general there has been much less focus on AID when compared to LID e.g. [196, 197].

3.8 Summary

In this chapter we have given an extensive review of specific models, techniques, challenges and variations of ideas to solve various problems that occur in the identification of gender, speakers, languages and accents from acoustic waveforms. Principles are borrowed across SID, LID, GID, AID etc. and we have given an overview that is both generic to all these problems, as well as very specific in some cases. The first sections of this literature review showed how past work in this field of research has successfully constructed a number of what we now consider to be standard approaches. We later on described the most recent developments for state-of-the-art techniques based on compensating for within-class differences, as well as channel differences. These techniques were first developed for SID but they have been very recently been transformed to the LID domain. As the later chapters of this thesis will show, we test these compensation techniques for the AID field, as well as propose some additions to the classification mechanisms based on how the i-Vector model behaves in the AID domain. Finally we gave an overview of the potential use and challenges for supra-segment, prosodic segments for the task of LID, and we use this as inspiration for some of our trials in AID, as we will show later in this thesis.

Chapter 4

Corpora

A number of different acoustic classification problems are considered in this thesis for GID and AID. For the evaluation of our experiments, a number of speech corpora were used. This chapter gives an overview of these datasets.

4.1 The TIMIT Acoustic-Phonetic Continuous Speech Corpus

The DARPA TIMIT Acoustic-Phonetic Continuous Speech Corpus [198] consists of read speech from 640 subjects spread across eight major dialects of American English. The sentences are chosen to be phonetically rich. There are 438 male subjects and 192 female subjects. Each subject reads 10 sentences, for a total of 6300 spoken sentences.

The dialect region is chosen by the geographical area of the United States where the speakers have lived during their childhood years. The breakdown is given in Table 4.1.

Table 4.1: Dialect regions represented in the TIMIT Corpus.

TIMIT Code	Region	#Male	#Female	Total
dr1	New England	31 (65%)	18 (27%)	49 (8%)
dr2	Northern	71 (70%)	31 (30%)	102 (16%)
dr3	North Midland	79 (67%)	23 (23%)	102 (16%)
dr4	South Midland	69 (69%)	31 (31%)	100 (16%)
dr5	Southern	62 (63%)	36 (37%)	98 (16%)
dr6	New York City	30 (65%)	16 (35%)	46 (7%)
dr7	Western	74 (74%)	26 (26%)	100 (16%)
dr8	Army Brat (moved around)	22 (67%)	11 (33%)	33 (5%)
1-8	all regions	438 (70%)	192 (30%)	630 (100%)

The text material in the TIMIT corpus prompts consists of 2 dialect sentences designed at the Stanford Research Institute (SRI), 450 phonetically-compact sentences designed at the Massachusetts Institute of Technology (MIT), and 1890 phonetically-diverse sentences selected at Texas Instruments (TI). The dialect sentences (the SA sentences) were meant to expose the dialectal variants of the speakers and were read by all 630 speakers. The phonetically-compact sentences were designed to provide a good coverage of pairs of phones, with extra occurrences of phonetic contexts thought to be either difficult or of particular interest. Each speaker read five of these sentences (the SX sentences) and each text was spoken by seven different speakers. The phonetically-diverse sentences (the SI sentences) were selected from existing text sources — the Brown Corpus and the Playwrights Dialog - so as to add diversity in sentence types and phonetic contexts. The selection criteria maximized the variety of allophonic contexts found in the texts. Each speaker read 3 of these sentences, with each sentence being read only by a single speaker. A breakdown of the speech material is given in Table 4.2.

Table 4.2: TIMIT material breakdown.

Sentence Type	#Sentence	#Speakers	Total	#Sentences/Speaker
Dialect (SA)	2	630	1260	2
Compact (SX)	450	7	3150	5
Diverse (SI)	1890	1	1890	3
Total	2342		6300	10

The TIMIT recordings were made using a close-talking noise-cancelling head-mounted microphones, and sampled at 16kHz.

4.2 The Accents of the British Isles (ABI-1) Corpus

The Accents of the British Isles (ABI) speech corpus [199] represents 13 different regional accents of the British Isles, and standard (southern) British English (sse). It was recorded on location in the 13 regions listed in Table 4.3 and contains speech from 285 subjects.

Each subject read twenty prompt texts, ranging from "task oriented" texts which are representative of generic applications of automatic speech recognition, to "phonetic" texts chosen for their phonetic content. The latter are the data used for our experiments.

For every accent group, twenty people were recorded (ten women and ten men) who were born in the region and had lived there for all of their lives. The standard southern English speakers were selected by a phonetician. Each of the 285 subjects read a set of 20 prompt texts,

Table 4.3: Accents represented in the ABI Corpus.

ABI Code	Location	Broad accent
brm	Birmingham	North, Midlands
crn	Truro, Cornwall	South, South West
ean	Lowestoft, East Anglia	South, East Anglia
eyk	Hull, East Yorkshire	North, Mid-North
gla	Glasgow, Scotland	Scotland
ilo	Inner London	South, London
lan	Burnley, Lancashire	North, Mid-North
lvp	Liverpool, NW Eng.	North, Mid-North
ncl	Newcastle, Tyneside	North, Far-North
nwa	Denbigh, N Wales	Wales
roi	Dublin, Ulster	Ireland
shl	Elgin, Scottish Highlands	Scotland
sse	Standard Southern English	South
uls	Belfast, Ulster	Ireland

which were divided in two categories of short or long phrases. The short phrases included:

- game commands e.g. “change view”, “select left”
- selected words that elicit specific vowel sounds e.g. “hide”, “hoid” (rhyming with “void”), “hoed” (rhyming with “showed”),“howd” (rhyming with “loud”)
- letters and international radio operator’s alphabet e.g. “G P Y O”, “yankee”, ”oscar”)
- digit sequences e.g. “four zero nine one”
- short phrases e.g. “while we were away”, “has a watch”

The long phrases contained:

- Equipment control commands e.g. “navigation select route home”
- SCRIBE sentences (British English version of the TIMIT sentences) e.g. “I itemise all accounts in my agency”
- An accent diagnostic passage

The ABI-1 recordings were made using head mounted and desk microphones, and sampled at 22.05kHz. The microphones used across the recording locations were the same. For the accent recognition experiments reported here, we are interested in the long accent diagnostic passage (we refer to as the sailor passage), which is split into three parts of approximately equal

length. We refer to these passages as 'SPA', 'SPB' and 'SPC'. The respective lengths are 92, 92 and 107 words, and have average durations of 43.2s, 48.1s and 53.4s. Only the head-mounted microphone recordings were used.

4.3 The WSJCAM0 Corpus

The WSJCAM0 [200] was recorded at the University of Cambridge and is the British English equivalent of a subset of the US American English Wall Street Journal corpus. It consists of speaker-independent read material, split into various sets that are usually used for training and testing in speech recognition applications.

The first set contains 90 utterances from each of 92 speakers. These sentences were taken from a subset of the WSJ0 training set of around 10,000 sentences, selected randomly. The same sentences could occur across different speakers, but never for the same speaker.

The second set contains 80 utterances from each of 48 speakers. Out of these, 40 of the utterances contain only words from a fixed 5,000 word vocabulary. The other 40 utterances are from a 64,000 word vocabulary.

The third set contains 18 adaptation utterances from all 140 speakers. These utterances include a single 3-second recording of background noise, two phonetically balanced sentences and the first 15 of the 40 sentences used for adaptation in the original WSJ0 corpus. These sentences were randomly selected and each sentence was allowed to occur in only one speaker's prompt material. No sentence repetition between or within speakers was allowed for this portion of the corpus. A breakdown of the WSJCAM0 corpus is given in Table 4.4.

Table 4.4: WSJCAM0 material breakdown.

Dataset	#Utterances	#Speakers	Selection Material
A	90	92	10,000 sentences
B(1)	40	48	5,000 words
B(2)	40	48	64,000 words
C	18	140	all

All recordings were made from two microphones: a far-field desk microphone and a head-mounted close-talking microphone, and sampled at 16kHz.

4.4 Summary

This chapter gave an overview of all the speech corpora for the various experimentation and analysis in this thesis. The TIMIT corpus consists of recordings collected across speakers from eight US dialect regions. All this material was used in our experiments. The ABI-1 corpus consists of read speech recordings collected from 13 different regions in the British Isles plus standard British English. Our work uses only the short passage utterances for our regional accent recognition experiments. This limits the amount of data available for having separate training, development and test sets. We shall later on describe how this dataset can be used in a "jack-knife" fashion, where we subdivide the data into three speaker-independent sets to overcome this limitation. Finally, the WSJCAM0 corpus consists of another set of recordings with various vocabulary limitations on different subsets of the corpus.

Table 4.5: A summary of speech corpora used in this thesis.

Corpus	Style	Channel	Sample Rate	Speakers	Utterances	Use
TIMIT	Read	Head mic	16kHz	640	6300	GID
ABI-1	Read	Head mic	22.05kHz	285	855	GID + AID
WSJCAM0	Read	Head mic	16kHz	125	625	GID

An overview of these corpora is given in Table 4.5. The ABI-1 corpus alone was specifically used for the AID classification experiments. For GID classification experiments, all the corpora described above are utilized.

Chapter 5

Gender Identification from Speech

Gender classification is useful in automatic speech systems. It is generally reported that having a good gender classification method is useful to separate models used for speaker identification or accent identification into gender dependent models [201, 202]. Gender accounts for a significant proportion of general speaker variability, and hence gender ID enables speaker identification systems to prune out a large number of speakers in a speaker classification system. Over an equally distributed population of speakers, perfect gender classification will localize the problem to 50% of the population at each test. In this chapter we see how the most popular acoustic correlate of gender differences in voice (pitch) is tricky to define over a large population. We then propose modelling pitch that is specific to a particular acoustic-phonetic context. Furthermore, we provide a feedback mechanism for "ambiguous" voices based on real-time pitch-shifting of the speech signal, which looks at how close/far the speech signal is from "unambiguous" gender training data, thus enabling a final decision to be made.

5.1 Gender and Pitch

One of the most pertinent features reported as useful for discriminating speaker voices is the fundamental frequency of the voice. Generally, typical values of *F*0 for male voices lie in a lower range than that for female voices. A demonstration of this is shown in Figure 5.1. The analysis was done on 6300 utterances from the TIMIT corpus (with 4380 male utterances and 1920 female utterances). The F0 pitch values were obtained using the algorithm by Talkin [51].

There are a number of observations we can make. The results obtained show that there

are two distinct pitch distributions for male and female pitch values. What is perceived to be a low pitch for males and a high pitch for females is a general trend in pitch values for these population subsets. The mean values (shown by dashed lines) might be taken to be the perceived zone of where we expect the pitch values of the respective genders fall. However, we can also notice that we should also expect some overlap, where high pitched male voices and low pitched female voices can crossover.

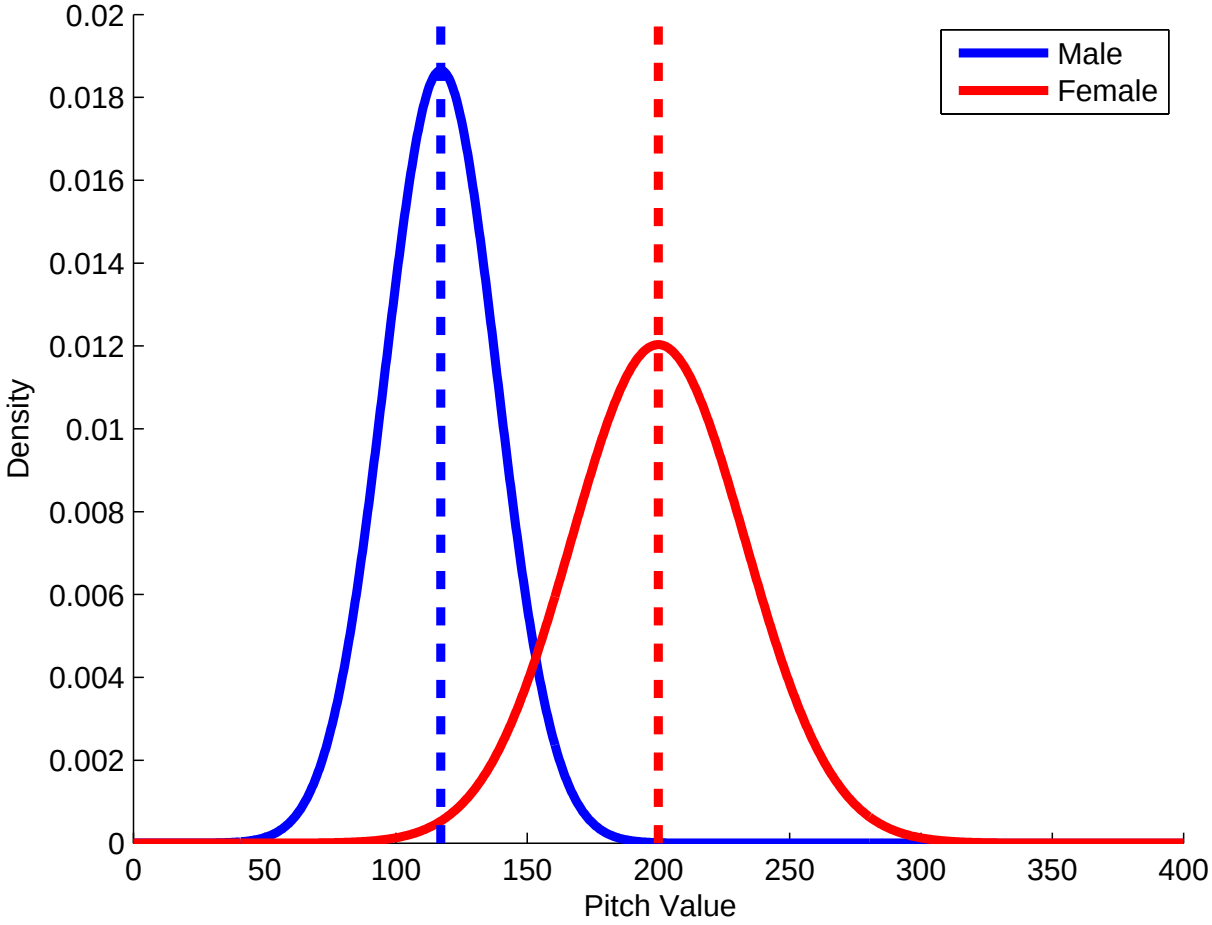


Figure 5.1: The probability density estimate of *F*0 values for the the TIMIT corpus with a kernel density estimator based on a normal kernel function.

We would like to perform a more detailed analysis to explain the factors that cause the overlap observed. We can hypothesise that, as well as the inherent range of pitch across different speakers of the same gender, age might be a factor, or perhaps, a particular phoneme of a language can exhibit its own range of pitch values within a gender.

5.2 Discovering Context

If we have a look at the same data, and this time account for speaker differences, by plotting a distribution based on individual speaker data, we obtain the results in Figure 5.2. The general distribution of both genders is equivalent to that in Figure 5.1. However, if we look at the distributions of individual speakers, we notice that the variance of each speaker is confined to a small fraction of the variance of the entire gender population. Female speakers exhibit wider variances on average than their male counterparts. This trend is also present in the distributions as observed in Figure 5.1.

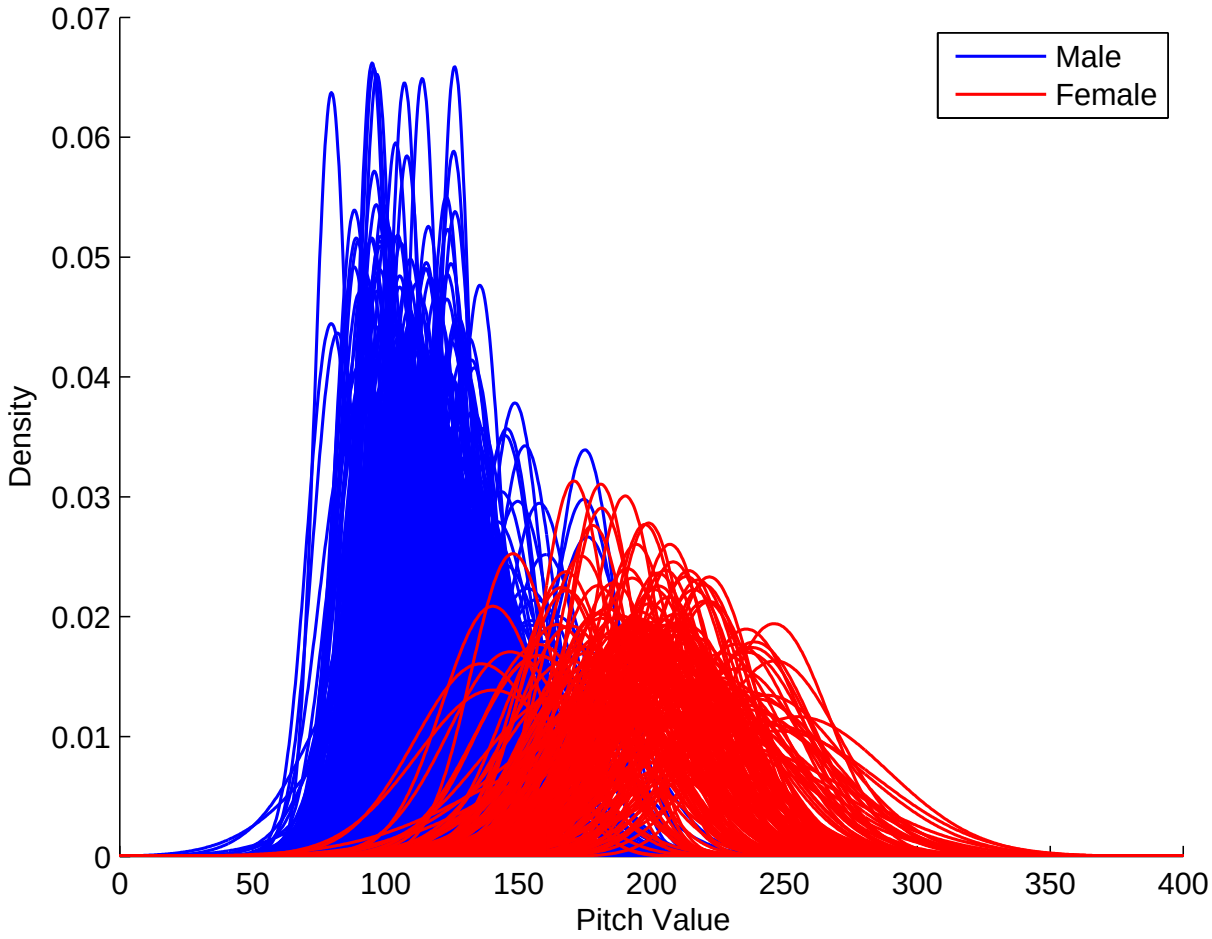


Figure 5.2: The distribution of *F*0 values for the speakers of the TIMIT corpus.

If we had prior knowledge of the speaker identity, then we could use that knowledge to make gender classification easier, as we would be able to model pitch behaviour for a particular speaker. The task we usually have, however, is to utilize information such as gender to identify a speaker, and not the other way round. We can look at the data from another perspective, that of phonetic context. Every frame is a unique vector of MFCCs, and it is not very useful to try and define the phonetic context by this information alone. Instead, we perform VQ over the

frames of the TIMIT corpus to obtain a codebook. Then, each MFCC frame is associated with the closest centroid in the codebook. This is what defines a more general phonetic context for a particular frame. Each frame has an associated pitch value. The algorithm used for pitch tracking is the one described by Talkin [51], and implemented in the 'Voicebox' toolkit [203]. We can now analyse at specific pitch distributions for each of the centroids in the codebook, for each gender. We show results for different numbers of context centroids in Figure 5.3.

There is not much variation in the behaviour of male pitch values as more centroids are added to the VQ codebook. At higher values of k, we can see two main pitch distributions of $F0$ values for male speakers. However, in the case of female speakers, VQ-based phonetic contexts show a multitude of varied pitch distributions within the main female pitch distribution observed initially in Figure 5.1. If the pitch of a speaker varies over a wide range of $F0$ values based not only on gender, but on the actual phonetic content that is being uttered, than $F0$ alone is a good, but not complete indication of gender. The production of a certain phonetic combination of sounds may require a lower/higher pitch relative to the actual perceived average pitch of a speaker (or entire gender group). But for the same phonetic combination by another speaker, also requiring a lower/higher pitch relative to another average pitch, the gender difference can be still detectable via these pitch values.

Even with just $k = 8$ and $k = 16$ context centroids defined, we can see how the distributions vary their mean considerably, despite some distributions having major regions of overlap. The male distributions appear to be more stable, and we can only observe two particular distributions, no matter what value of k is chosen. Modelling the actual pitch distributions for each phonetic context can be done with a low-order number of Gaussians, which would mostly be required by female speakers rather than male speakers.

It seems possible that an adequate number of contexts coupled with an adequate number of Gaussians to model each context can capture the variability in $F0$ for both genders under different acoustic contexts. For this reason, our approach [8] is designed to find a close acoustic context to the content that is being analyzed in speech using predefined acoustic templates built by MFCC codebooks. Once the acoustic context is determined, the pitch information expected within that acoustic context is compared to male and female pitch templates and a gender decision is made. Furthermore, we exploit inconsistencies between gender classifiers by looking at the effect of pitch-based distortions of the original speech signal to give a refined classification where possible.

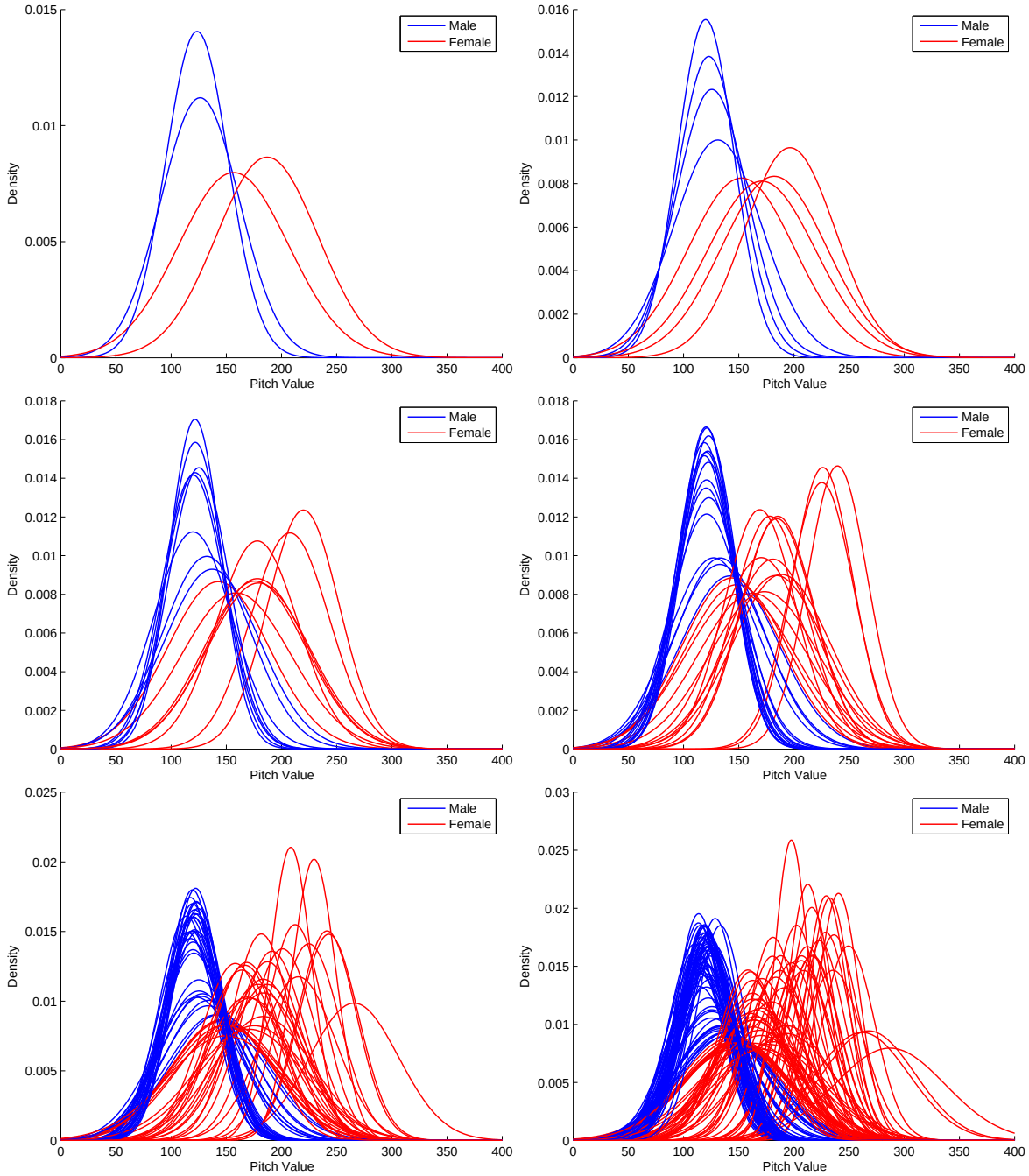


Figure 5.3: The distributions of *F*0 values across all speakers for different phonetic contexts (left to right, top to bottom: 2,4,8,16,32,64 contexts) of the TIMIT corpus.

5.3 Gender Classification Methodology

5.3.1 Baseline Classification

In our baseline classifier, we use MFCC feature vectors extracted from continuous speech, either from the TIMIT [198] or from the ABI [199] corpora. Two different vector quantizer models are built, one per gender, by clustering (K-Harmonic Means) the MFCC feature vectors from the training data of each gender. The MFCC vectors utilized had 12 coefficients, where the 0^{th} coefficient was excluded. To classify a test utterance $X = \{\mathbf{x_1}, \ldots, \mathbf{x_T}\}$ with the reference centroids $R = \{\mathbf{r_1}, \ldots, \mathbf{r_K}\}$ from the clustering, the standard average quantization distortion is calculated as in Equation (5.1), where $d(\cdot, \cdot)$ is the Euclidean distance $\| \mathbf{x_t} - \mathbf{r_k} \|$.

$$D_q(X, R) = \frac{1}{T} \sum_{t=1}^{T} \min_{1 \leq k \leq K} d(\mathbf{x_t}, \mathbf{r_k}) \tag{5.1}$$

Each frame from a test utterance is therefore associated with a particular centroid (the closest) from the VQ codebooks from the male and female training data. The distance/distortion is measured for all frames for both male and female codebooks. The codebook that gives the least over all distortion for the test utterance X is assumed to originate from the particular gender model that holds R with the smallest distortion.

5.3.2 Context-Dependent Classification

The centroid models provided by MFCC clustering give an unlabelled indication of where different units of sound lie in MFCC space. Rather than using these directly in a classifier, we construct Gaussian Mixture Models (GMMs) of the pitch values associated with each MFCC vector that was included in the calculation of the centroid. The motivation of this technique is that the MFCC centroid positions correspond to different contexts of sounds, and these contexts can affect the pitch produced. This is evident from various experiments of pitch distributions and ranges for various sounds, and combinations of sounds.

The architecture of this method is illustrated in Figure 5.4. Three frames from an utterance need to be classified. A prior codebook in MFCC space has been generated for each gender, with $k = 4$ centroids per gender. Each frame is associated with the closest centroid for each gender codebook. In this case, the MFCCs from the first frame are associated with centroid B (from the male codebook) and C (from the female codebook), the MFCCs from the second frame

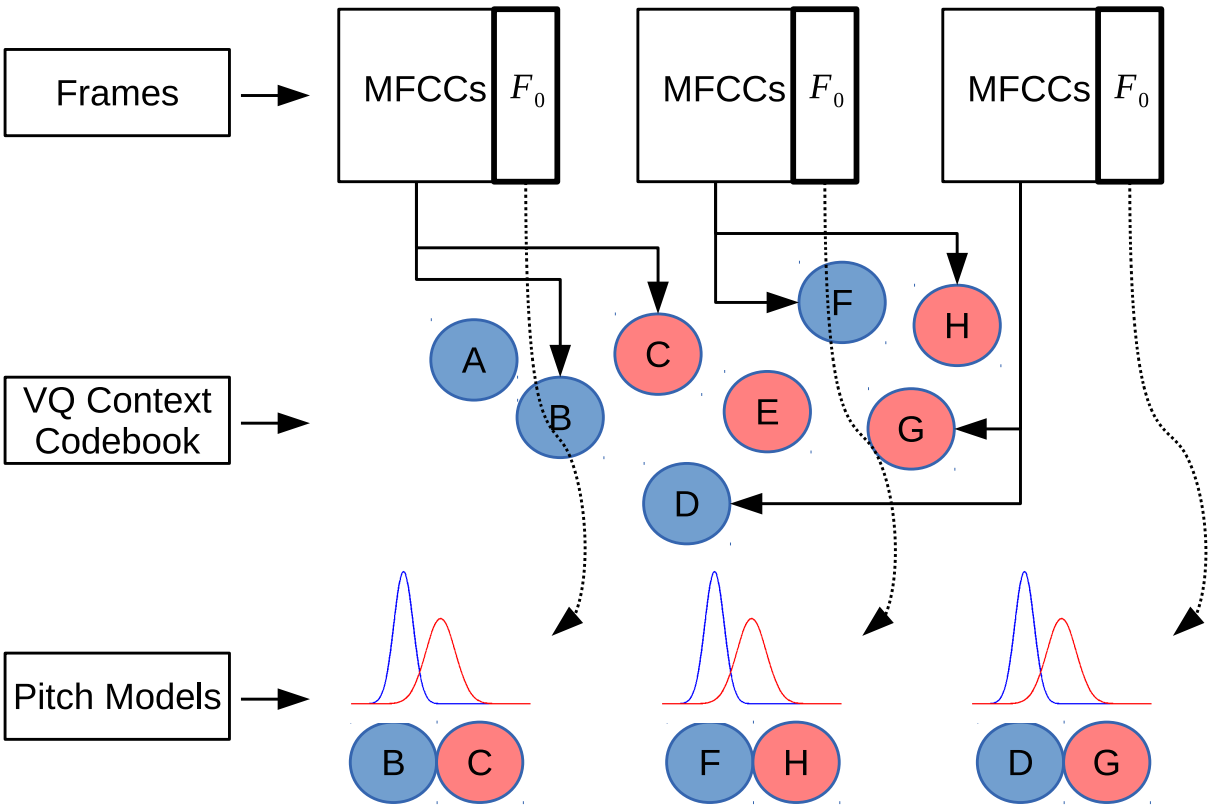


Figure 5.4: Gender classification is based on the acoustic context that is associated with a particular frame, and specific pitch models for that context only are used to classify a particular frame. In this example, the MFCCs for the first frame are associated to centroid B (from the male VQ codebook) and centroid C (from the female VQ codebook). Consequently, the F_0 value for the frame is scored under the pitch model for these selected centroids only. The same applies for every frame in the utterance.

are associated with centroid F (from the male codebook) and H (from the female codebook), and the MFCCs from the third frame are associated with centroid D (from the male codebook) and G (from the female codebook). Each centroid from the codebook has a specific pitch GMM associated with it as discussed earlier, and these GMMs are used to calculate the log likelihood of the pitch for the frame under the GMM for both the male and female distributions for the contexts in question. For the whole utterance, these likelihoods are summed to check which overall gender model gives the best fit for the observed utterance.

5.3.3 Pitch-Shifting Loop-Back Classification

If both the classifiers described above in Section 5.3.1 and Section 5.3.2 give the same classification, then there is reasonable confidence that the classification result is correct and the gender is confirmed. However, if there is disagreement, an additional "acoustic loop-back" process is utilized.

Groen et. al [204] perform a number of experiments related to the human perception of gender in voice for children. Their interest was in investigating the difference in response time between children with high-functioning autism and normal children. The main finding of interest to us is that the response time for gender perception for both groups changes in specific cases, as the pitch of a voice is artificially transformed into subsequent pitch categories by shifting formant ratios and median-pitch levels, from male to female voices. This suggests that the brain process that classifies gender can have different cognitive loads in cases where gender determination is ambiguous. This observation motivates us to propose an extra layer of processing to resolve the classification in cases where the two classifiers disagree, which we take as an indication that the gender information is ambiguous. This processing can be visualized as measuring whether the ambiguous utterance is in fact closer to the male or the female gender in the pattern-space. We do this by small artificial pitch-shifts on the utterance in either the male or female direction, and then re-classifying it with the two classifiers described earlier, to see if they now agree.

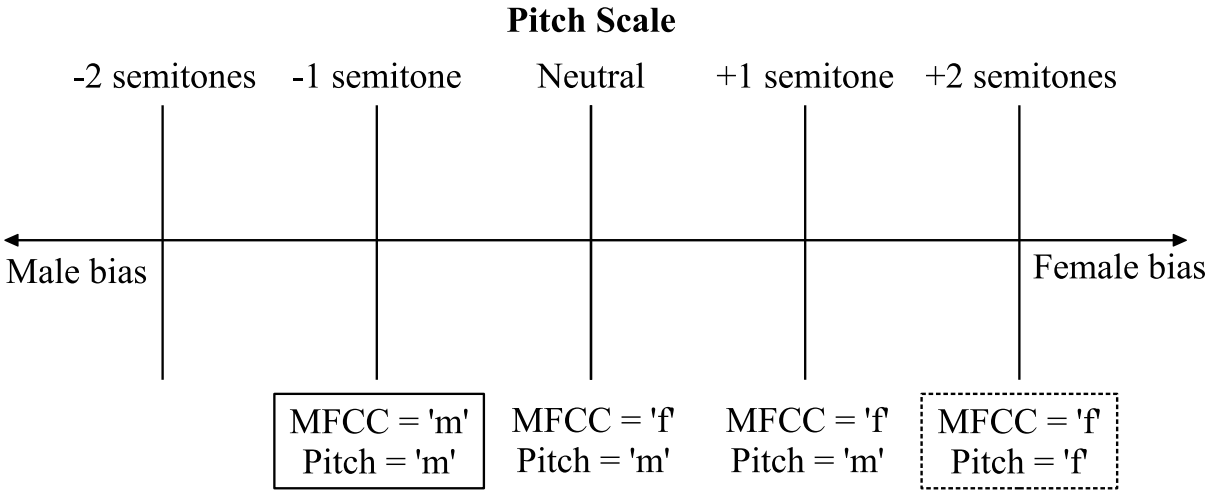


Figure 5.5: Pitch-shifting on an utterance. Shifting to the right from the neutral position implies an upward shift of pitch towards the female gender, shifting to the left implies a downwards shift towards the male gender. After each semitone shift, the decisions of the two classifiers ('MFCC' and 'Pitch') are shown. Agreement on the gender is reached after only one semitone shift downwards, but after two semitones upwards, so the utterance is classified as a 'male'.

The process works as follows: in cases where there is a disagreement in the classification results from the two classifiers, two copies of the utterance are made. Copy 1 is shifted downwards progressively in pitch steps of a semitone, and Copy 2 is shifted upwards progressively in pitch steps of a semitone, to a maximum shift of two semitones. After the first shift, the utterance is re-classified by the two classifiers. If the classifiers now agree on one gender (only), this gender is taken as the class, and the process ends. If not, another shift is applied. Figure 5.5 gives an example. With no pitch-shift applied, the two classifiers disagree. Agreement is reached

between the classifiers in two situations: either when the pitch is shifted downwards by one semitone, or when shifted upwards by two semitones. Because the utterance requires only one semitone shift downwards to make the classifiers agree on 'male', then this gender is taken as the correct class.

The process of upwards/downwards pitch-shifting and reclassification is iterated until one of the following exit conditions is met:

- The classifiers agree on the class 'male' after a downwards pitch-shift, and this shift is smaller than the last upwards pitch-shift, after which they still disagreed. In this case, the gender 'male' is chosen.
- The classifiers agree on the class 'female' after an upwards pitch-shift, and this shift is smaller than the last downwards pitch-shift, after which they still disagreed. In this case, the gender 'female' is chosen.
- The classifiers agree on the class 'male' after a downwards pitch-shift of two semitones, and on the class 'female' after an upwards pitch-shift of two semitones. In this case, the classification made by the acoustic context classifier result is used.
- The pitch has been shifted by two semitones in both directions and the classifiers still disagree. In this case, the classification made by the acoustic context classifier result is used. This is because the classifier in Section 5.3.2 is generally more accurate than the one in Section 5.3.1.

Pitch-shifting is done using the 'SoundTouch' audio processing library [205].

5.4 Experiments

A number of experiments were performed on the TIMIT [198], ABI-1 [199] and WSJCAM0 [200] corpora. The TIMIT corpus contains 438 male speakers and 192 female speakers, where each speaker speaks 10 phonetically rich short utterances. The ABI-1 corpus subset used contained 145 male speakers and 140 female speakers (chosen to balance the number of speakers in each gender), where each speaker speaks 3 extracts of 6 seconds each from accent diagnostic passages. The WSJCAM0 corpus subset contained 55 female speakers and 70 male speakers, where each speaker speaks 5 utterances of around 3-5 seconds each.

For every experiment we conduct, 100 male and 100 female speakers are selected to train the gender models. Only the TIMIT and ABI-1 corpora were used for training, at various stages of experimentation. The WSJCAM0 was not included as a training set. This ensured that at least one corpus was presented to the tested algorithms as a completely new dataset. The speakers chosen for training were consequently not utilised for testing, which was done on the remaining set of speakers. The entire set of experiments were conducted with 5-fold cross-validation, with results pooled together for the GID accuracy rate.

We used four different experimental training and testing data pairs. TIMIT/TIMIT is a classification of TIMIT data based on training over TIMIT data. ABI/ABI is a classification of ABI-1 data based on training over ABI-1 data. TIMIT/ABI is a classification of ABI-1 data based on training over TIMIT data. TIMIT/WSJCAM0 is a classification of WSJCAM0 data based on training over TIMIT data. In TIMIT/ABI and TIMIT/WSJCAM0 experiments, training data was collected from 100 male and 100 female TIMIT speakers, whilst tests were performed on all the ABI-1 and WSJCAM0 speakers.

The frontend processing is common to all experiments. Utterances are framed in 30ms segments with a 15ms frame rate. Each frame is represented by a 12-dimensional MFCC vector, where the 0^{th} coefficient was excluded. A pitch value for each frame was estimated with the pitch tracker algorithm by Talkin [51]. All unvoiced frames (where no pitch is present) are discarded. No frame normalization techniques like CMS or feature warping were applied at any stage.

5.4.1 Matched Dataset Tests

For matched dataset tests, we consider the cases where the testing utterances are from the same corpus as those used for training, albeit different speakers. The results for TIMIT/TIMIT experiments are shown in Figure 5.6, whilst the results for ABI/ABI experiments are shown in Figure 5.7.

The results for TIMIT/TIMIT tests show that the MFCC classifier performance improves gradually as the value of k (number of cluster centroids) increases from 2 to 16. At this point a performance barrier is reached, and no improvement can be seen at higher values of k. However, the context-dependent classification as well as the pitch-shifting loopback classification maintain a steady performance across all values of k. The variance in the results obtained by the context-based classifier and the pitch-shifting loopback classifier we are proposing in this paper shows

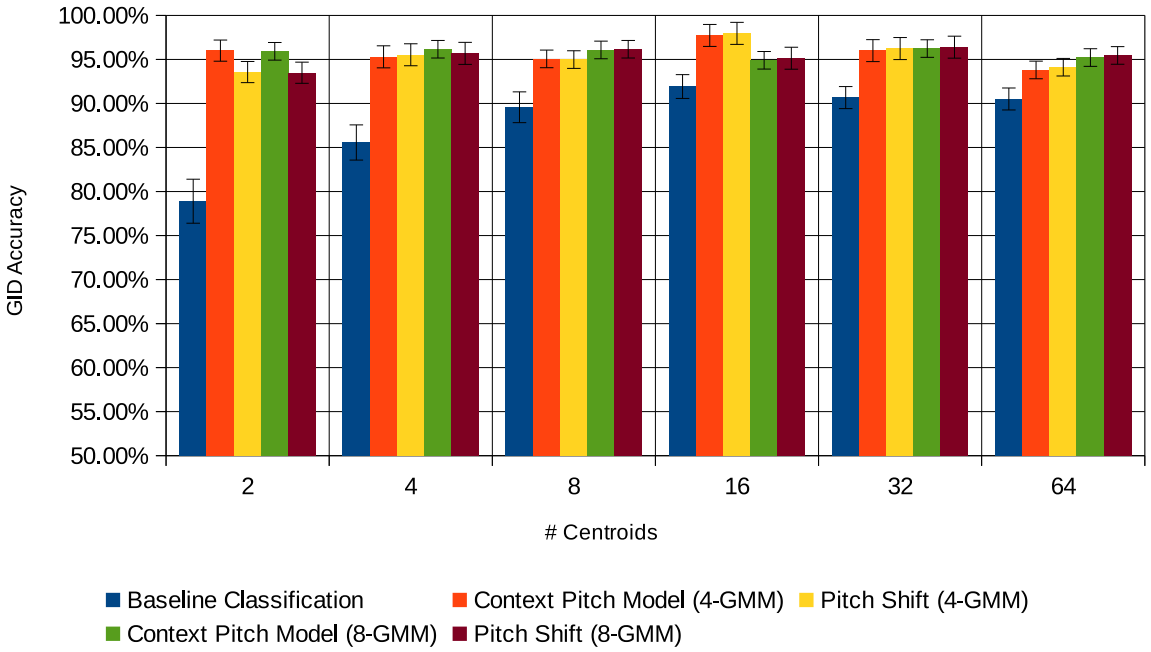


Figure 5.6: GID accuracy for TIMIT/TIMIT experiments.

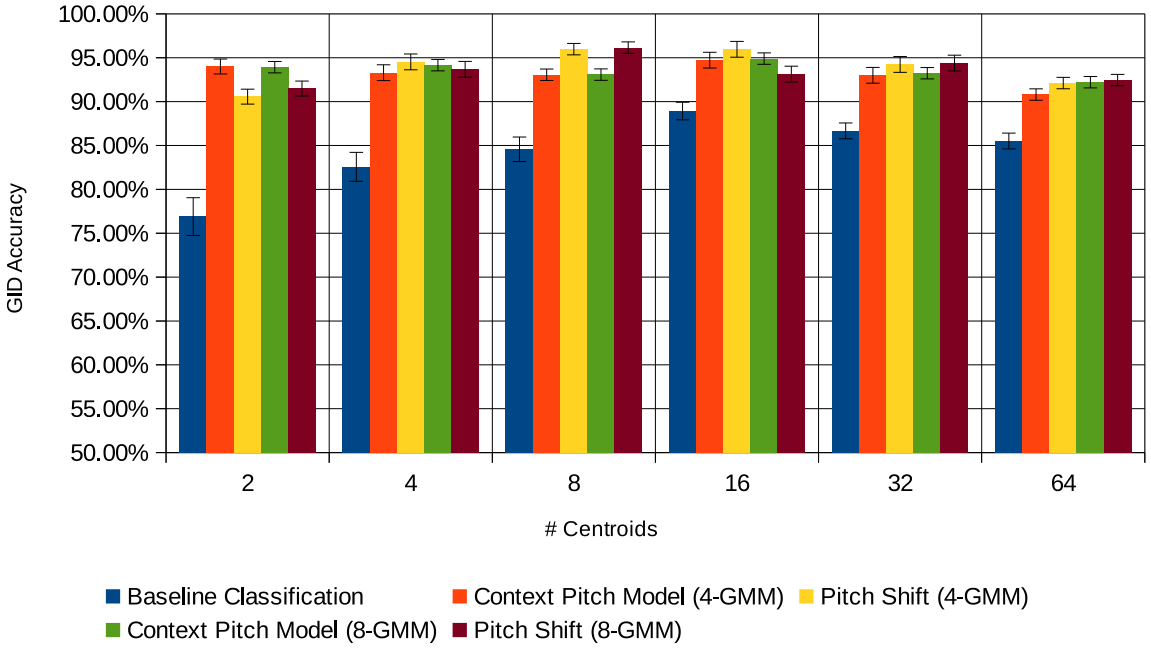


Figure 5.7: GID accuracy for ABI/ABI experiments.

greater stability compared to the baseline MFCC classifier, as well as giving higher gender identification performance across all experiments. On the other hand, there is no regular gain observed for the pitch-shifting loopback classifier, which performs better or worse depending on the value of k.

The results for ABI/ABI tests show that globally, identification results on all classifiers

perform slightly worse on the ABI-1 corpus, when compared to performance on the TIMIT corpus. However both the context-based classifier and the pitch-shifting loopback classifier still perform better than the MFCC classifier. The MFCC classifier performance improves gradually as the value of k (number of cluster centroids) increases from 2 to 16, and drops for $k > 16$. The overall drop in performance on the MFCC classifier (compared to TIMIT/TIMIT tests) is associated with a drop in performance in the other classifiers. Again, the pitch-shifting loopback classifier does not always perform better than the context-based classifier.

5.4.2 Mismatched Dataset Tests

For mismatched dataset tests, we consider the cases where the testing utterances are from a different corpus from those used for training. The results for TIMIT/ABI experiments are shown in Figure 5.8, whilst the results for TIMIT/WSJCAM0 experiments are shown in Figure 5.9.

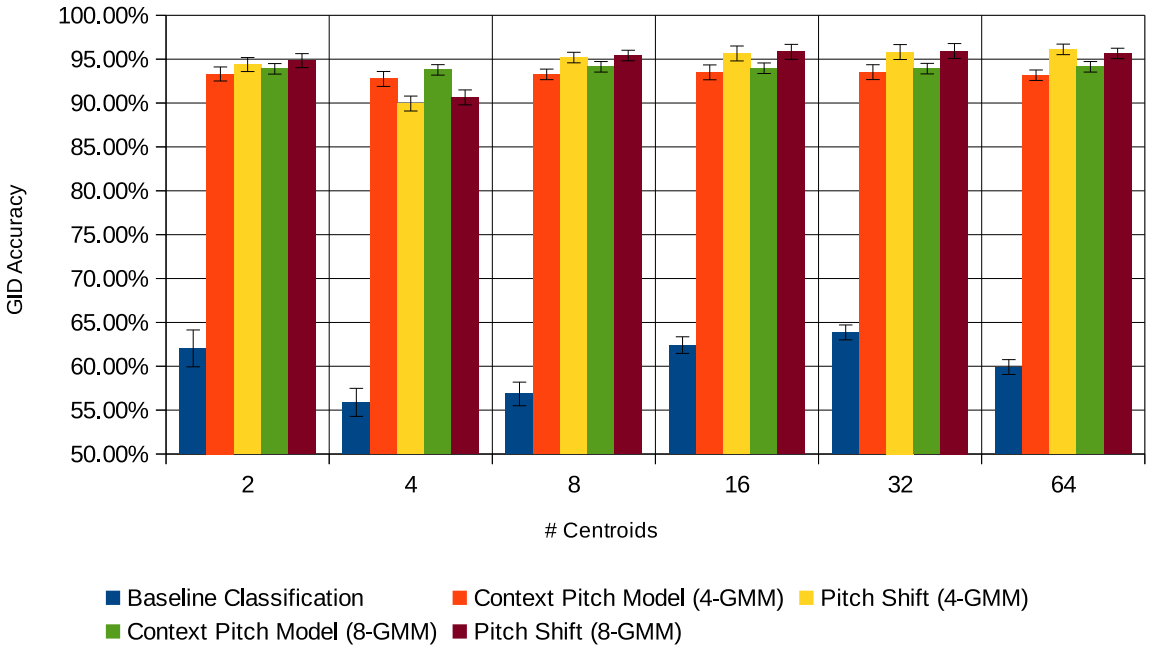


Figure 5.8: GID accuracy for TIMIT/ABI experiments.

The results for TIMIT/ABI tests show that the baseline classifier accuracy is lower than in cases where the training and testing sets are the same, and is much lower when training is performed on TIMIT and testing on ABI-1. This indicates that the method is not very robust for classification on different training/testing data sets. The drop in the MFCC classifier is of approximately 30%. On the other hand the context-based classifier maintains a very high and stable classification score in the range of 93-94% accuracy. The pitch-shifting loopback classifier further boosts the results in almost all cases, with a stable result of 95% region for values of

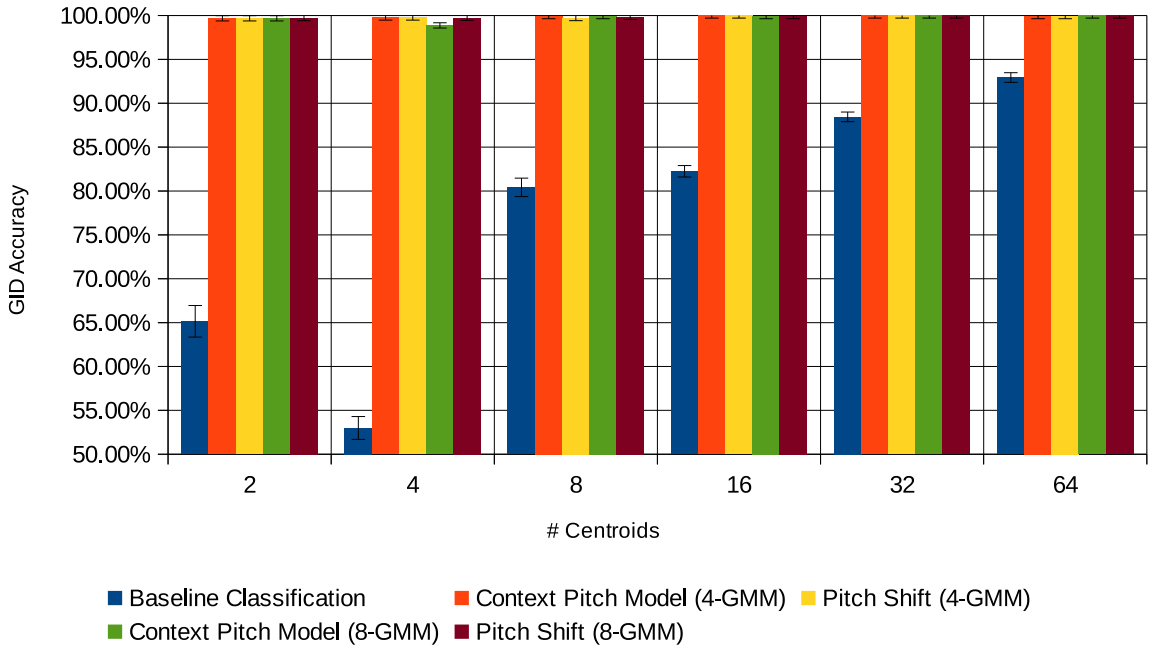


Figure 5.9: GID accuracy for TIMIT/WSJCAM0 experiments.

$k > 8$, which is very close to the classification accuracy obtained in TIMIT/TIMIT and ABI/ABI tests. The gain for this extra classification stage is therefore greater in TIMIT/ABI tests, and the conclusion is that it is reconciling many errors that occur due to unmatched training/testing data sets.

The results for TIMIT/WSJCAM0 tests show that the baseline classifier starts very poorly, in a similar way to TIMIT/ABI tests. The performance improves at higher values of k. However, the performance of the acoustic context and pitch-shifting classifiers is very high on all values of k, further again demonstrating the gain these algorithms have on mismatched training/testing sets. To note is that GID performance at or close to 100% accuracy was achieved in some of these tests. This suggests that the WSJCAM0 corpus presents a particularly easy speaker set for GID. The GID task has had ample research and is not considered to be a hard problem. The experiments we report here are meant to look at the differences (all else being equal) of different algorithms as datasets are shifted from training to testing.

5.4.3 Pitch-Shifting Utilization

The relative number of male and female utterances classified without pitch-shifting and using 1 or 2 semitone shifts is shown in Table 5.1. Analysis of results shows that female utterances require the intervention of the pitch-shifting process earlier than male utterances, and in the

greater majority of cases require two pitch-shifts before classifiers could agree on gender classification. Also, gender-identification on female speech utterances require more intense use of pitch-shifting (2 shifts) than in male speech utterances. This corroborates the experimental results by Groen et. al [204], which concluded that humans take longer to classify gender for female speakers when it is ambiguous than they do for male speakers, and secondly, that more female utterances than male utterances sound ambiguous in pitch. Also, if we refer back to the analysis leading to the results in Figure 5.3, where the female speakers pitch values had very distinguishable sub-regions as more phonetic contexts were defined. Quite a large number of these contexts for female speakers overlapped considerably with the pitch distributions for male speakers, which however, have a much more stable distribution no matter how many contexts are defined.

Table 5.1: Pitch-shifting utilization across utterance for male and female speakers. The columns show the relative number of tested utterances that required no pitch shift, one pitch shift or two pitch shifts respectively.

	% 0 shifts	% 1 shift	% 2 shifts
Males	65.68%	24.55%	9.77%
Females	46.43%	16.47%	37.10%

The pitch-shifting classifier, in general gives some improved results over the context-based pitch model classifications, especially for mismatched training and test sets. In the case of equivalent training and test sets the pitch-shifting classifier can actually give slightly worse performance. So whilst these results provide some analytical interest, the differences in performance obtained by pitch-shifting over context-based classification are not that statistically significant, and we cannot guarantee that the results obtained in these experiments would apply in all conditions.

5.5 Summary

In this chapter we have first looked at the behaviour of the acoustic correlate of "gender" in speech, most often interpreted to be pitch. Pitch is of course not exclusive, and there are other useful correlates one could investigate, such as formants. We have seen how the behaviour of pitch differs to quite an extent between male and female populations. The female population has considerably more pitch variability for different acoustic contexts when compared to the male population. Because of this, we have constructed a representation where the phonetic context divides the pitch values for each gender into specific submodels (GMMs of low order).

Each voice frame extracted from speech can be analysed in a more local way, and the results of an entire utterance can be pooled together for a final decision.

Furthermore, we have described a simple pitch-shifting process guided by classifier fusion, that gives a useful gain in gender identification performance, especially on unmatched training/testing sets. The behaviour of the pitch-shifting process loosely corroborates some of the observations made in experiments on cognitive load in humans for speech gender classification. It would be interesting to find other speech features that exhibit similar properties on warping/shifting. In some cases, the upper bound of the accuracy of the MFCC classifier is holding down the potential of the context-based classifier. Therefore a replacement of MFCC features with a feature set that is more gender-specific, rather than speaker-specific, could boost the results of the techniques presented here.

Chapter 6

Acoustic Accent Identification

The next chapters of this thesis will focus on the problem of accent identification from speech. In this particular chapter we will give an overview of a first set of experiments, based on a number of standard generative and discriminative classifiers that usually perform well on other identification problems such as speaker and language identification. However, the performance obtained for accent identification was sometimes very poor and overall, disappointing — thus setting accent identification apart from these other problems. Whilst these standard techniques have been applied to accent identification in previous work, they have been combined with some form of frame or model based speaker normalization techniques, which we do not apply in this chapter. The next chapters will consider the problem of speaker variability modelling and how this effect can be attenuated. In order to compare the differences obtained with speaker variability modelling, it is important to have a baseline starting point, which we develop in this chapter. This chapter will attempt to demonstrate how, without such forms of compensation, the information that characterises accents is obscured somewhat by speaker variability. The experiments in this chapter are a sequence of "approaches", which build on the insight gained from the previous approach to solve the accent identification problem, and build up to a moderately good classification system by the end of the chapter.

6.1 GMM-UBM (Approach I)

The GMM-UBM classification system is popular in speaker and language identification research. It is a simple and effective modelling and classification solution that gives good performance in the SID and LID domains. Given short-term feature vectors estimated from speech utterances,

describing spectral information, a class can be modelled, and the models should differ enough in their spectral content to be able to infer reliable classifications of speakers or language. This first experiment is to assess whether the raw spectral information extracted from utterances are good enough for the accent identification problem.

6.1.1 Feature Extraction

Every utterance is framed into segments of 30ms with a 15ms frame rate. Voice activity detection (an implementation of [206]) is performed to remove silent portions of the signal. MFCCs are extracted from the utterance: 13 coefficients over 30 filters that range over the full bandwidth of the utterances (11025 Hz). The MFCCs are then converted to shifted delta cepstra (SDCs) with a 7-1-3-7 parametrization (see Section 2.3.5). The original MFCCs are then Gaussian-warped over three second time windows(see Section 2.3.6.3). The warped MFCCs are then concatenated to the SDCs to form the final feature vector. Each frame is therefore described by a 62-dimensional feature vector.

6.1.2 GMM Modelling

The first trial is designed to set a baseline for AID. The experiment is based on the GMM-UBM classification technique. The GMM is a sound statistical model. The acoustic features under a GMM are modelled without any reference to their position within a particular linguistic unit or the prosodic style that generated it. The assumption is that any frame sequence extracted from an utterance is statistically independent from all other frames in the sequence. The second assumption is that the acoustic features extracted are distributed over some mixture of normal distributions and are uncorrelated (diagonal covariance matrices are used in the components).

As explained in Section 4.2, the ABI-1 corpus is split into three roughly equal sets speakers for each accent, where data from one speaker is found in one set alone, and in no other sets. Three test trials are performed. In each trial, two sets are used for training and one for testing, and the results are pooled for a final classification result. In the GMM-UBM experiment, a single UBM is created from the entire training section of the corpus. Consequently one GMM per accent is created by means only MAP-adaptation (with a relevance factor of 16) of the UBM to training data for each accent. We have to note that by doing so, UBM training for AID is different to UBM training in speaker recognition. In a speaker identification system, the UBM usually comes from a completely different set of speakers so that the eventual recognizer can 'enroll'

new speakers sequentially. Furthermore, at test stage, one cannot use other target speakers for normalization purposes. Of course, in the problem of accent identification, we expect the same accents in the training set as in the test set, albeit from different speakers. So no actual 'enrollment' occurs. The database conditions are therefore more static in AID compared to SID.

An important consideration in our UBM modelling is the estimation of parameters to define the UBM. Each mixture component in the UBM is defined by a mean, covariance matrix, and weight. Though there are methods that estimate a Gaussian Mixture Model given a dataset, we were noticing that since our corpus was not a large one, components tended to converge to singularities (this is when a mixture component collapses on a particular point, with the point becoming the mean, and the component has zero variance, with a likelihood approaching infinity) when training GMMs of large component sizes such as 512 and 1024. Therefore we opted for a stable way to estimate a GMM. Firstly, we obtain a VQ codebook via the Linde-Buzo-Gray (LBG) algorithm. The codebook splitting criteria we used was to double the number of centroids at every LBG iteration, and then the centroid means were re-estimated via traditional k-means algorithm until the desired number of centroids is reached. Once the cluster centroids (the VQ codes) are estimated, the covariances and weights of each cluster are estimated. This initial estimation is then passed on to a GMM trainer to perform five iterations of Expectation-Maximization, which outputs the final UBM. By using this method of GMM training as opposed to direct Expectation-Maximization on the dataset, we resolved all our singularity problems even on large component GMMs. Even so, given the rather small amount of data in the ABI-1 corpus, we limit all our models to a maximum of 1024 components, as larger GMMs (say 2048 or 4096) would require exponentially more data to create stable models.

6.1.3 Scoring

Given a particular set of frames for an utterance $X = \{\mathbf{x_1}, \mathbf{x_2}, \ldots, \mathbf{x_N}\}$, we can calculate the sum of log probabilities of the entire set of frames under a particular GMM λ_m with Equation 6.1. No normalization term is used here. We are dealing with a closed-set identification problem, so the normalization term is the same for all accent groups, and is therefore not necessary.

$$\text{Score}_m(X) = \frac{1}{N}\sum_{n=1}^{N} log\left(P(x_n|\lambda_m)\right) \tag{6.1}$$

Given that we are classifying 14 different accents, each utterance is scored under all 14 GMMs representing each accent. The model with the highest score for the utterance is chosen as

the classification result and the accent associated with that model is assigned to the utterance.

6.1.4 Results

The results for GMM-UBM classification on short-term feature vectors are shown in Figure 6.1. The results show a clear trend of improving as the number of components increase. The actual classification results are however, quite poor. Given 14 different accents, the chance level is at 7.14%. So there is definitely some measure of accent classification being executed correctly given the speaker-independence in our tests, ranging from 28% to 42% AID accuracy.

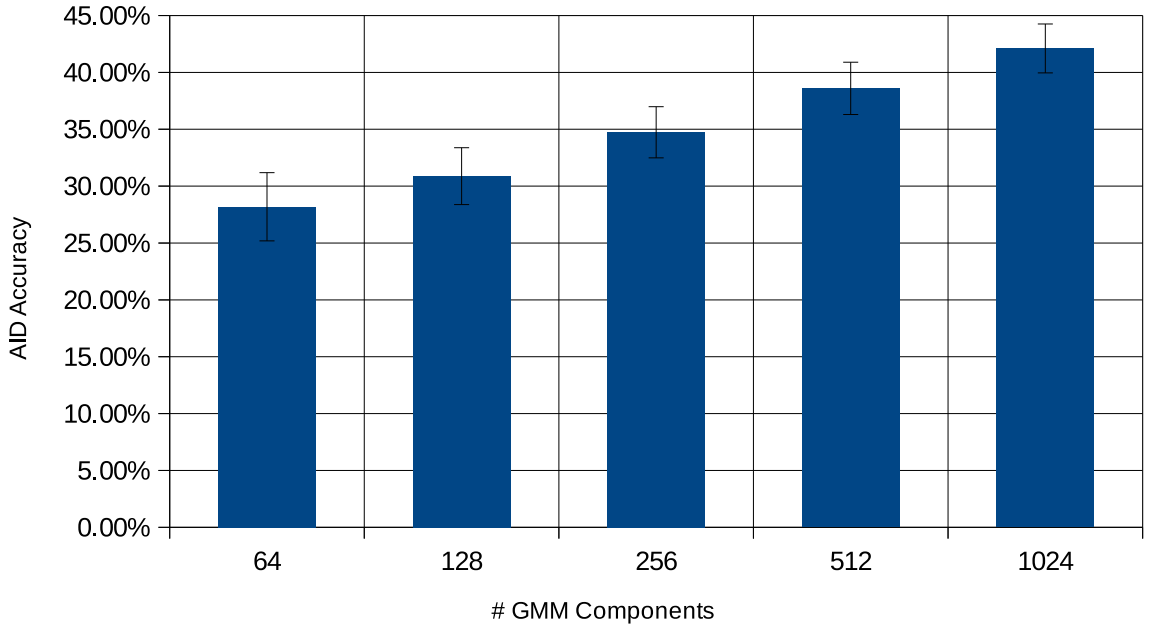


Figure 6.1: Accent identification results for Approach I: GMM-UBM classification on short-term feature vectors.

6.2 GMM-UBM with Prosody (Approach II)

In the second approach, the same GMM-UBM training and scoring system is used as in Approach I. The differences are in the feature vectors collected for an utterance.

6.2.1 Feature Extraction

Every utterance is framed into segments of 30ms with a 15ms frame rate. Voice activity detection is performed to remove silent portions of the signal. MFCCs are extracted from the utterance:

13 coefficients over 30 filters that range over the full bandwidth of the utterances (11025 Hz). The MFCCs are then Gaussian-warped over three second time windows. The MFCCs are then converted to shifted delta cepstra (SDCs) with a 7-1-3-7 parametrization. For each frame the pitch and first formant and bandwidth for the formant are also calculated. The first derivatives (or deltas) of the pitch values of the signal are also calculated. Unvoiced frames (those with no corresponding pitch value present) are discarded. The original MFCCs are not included, and therefore each frame is described by a 53-dimensional feature vector.

6.2.2 Results

The results for GMM-UBM classification on short-term feature vectors that included cepstral features as well as prosodic and intonational features are shown in Figure 6.2. Again, the results show a clear trend of improving as the number of components increase. The actual classification results, though still relatively poor, are better than the ones in Approach I for all the GMM component sizes tested. The main difference is that some importance to pitch information is given in this approach, whilst the standard MFCC information is discarded, as well as all unvoiced frames. The range of accent identification accuracy now stands in the range of 29% to 49%.

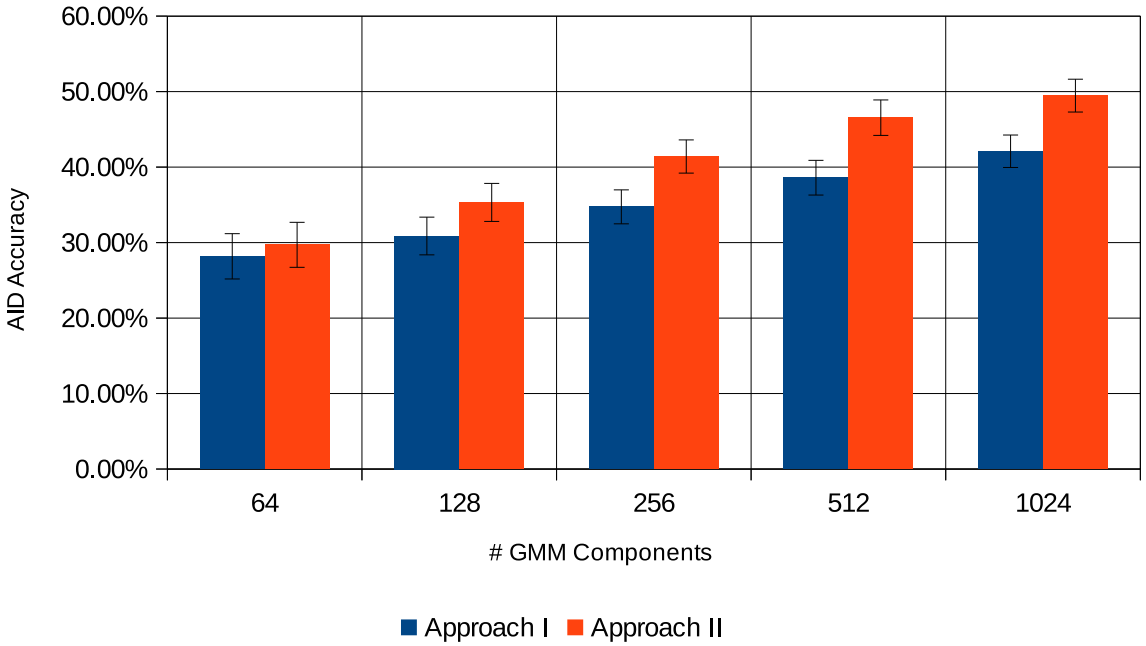


Figure 6.2: Accent identification results for Approach II: GMM-UBM classification on short-term feature vectors that include prosody and intonational information.

The results from this approach suggests that by adding some prosodic information with

dynamic cepstral information and removing the basic MFCC feature vector, we might create a frontend configuration that is more tuned to the accent identification problem.

6.3 GMM-UBM with Prosody Context (Approach III)

We require a way of segmenting speech signals into long-term syllabic-like acoustic segments which may be characteristic of different accents. Of course, this has to be done without an actual phone-recognition front-end. In this approach we split utterances into contiguous segments of 100 frames each, which at a 15ms frame rate, amount to 1.5s of speech data. These segments of speech are not overlapping. A few subjective listening tests seemed to suggest that 1.5s of speech contains, in some cases, prosodic structures that are quite telling of accents. However, we note that the duration of 1.5s was ultimately an arbitrary decision.

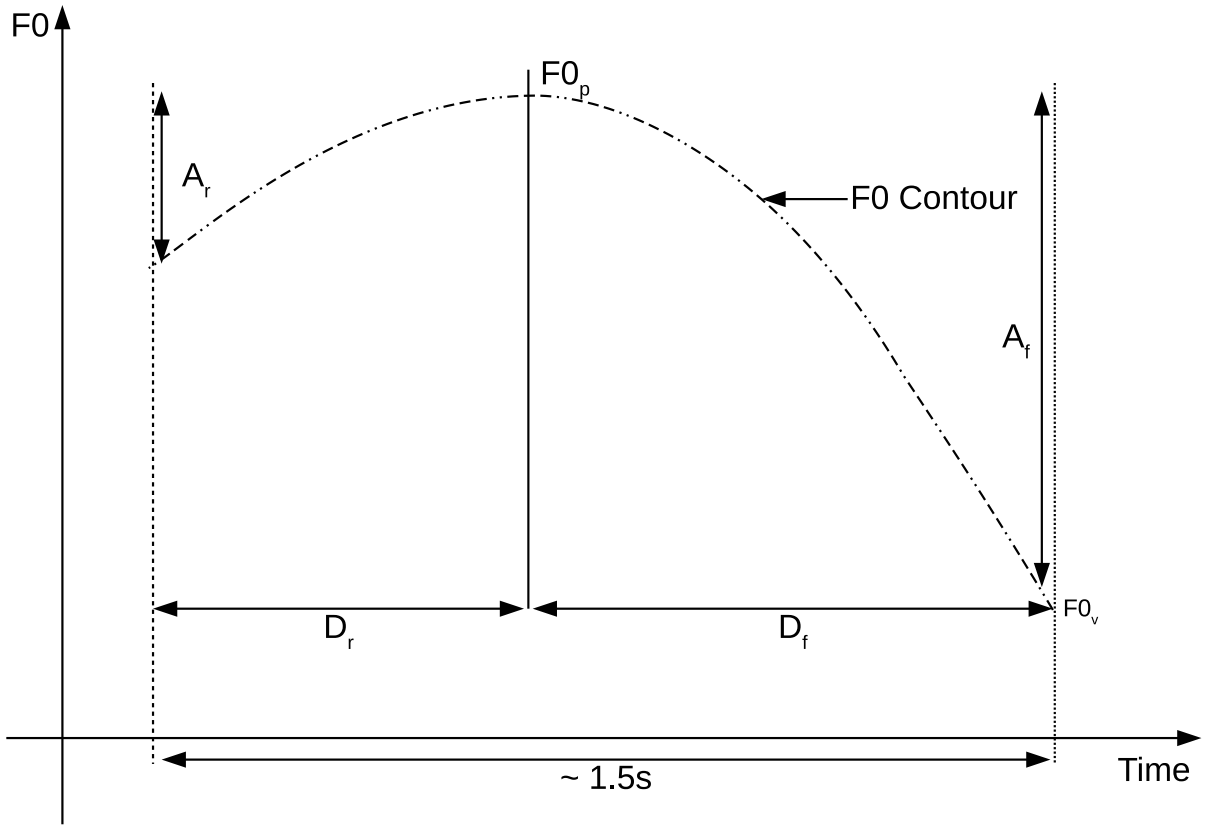


Figure 6.3: Tilt parameters to calculate pitch dynamics for a speech segment.

For this task, we want to capture prosodic information that can not only adequately represent, in a limited number of dimensions, the prosodic behaviour of an utterance, but also information that can be modelled in such a way as to show differences across accents. We assume that the dynamics of the pitch contour are roughly consistent for the same speaker speaking the same

utterance. The pitch contour is not only dependent on the speaker, but also on the underlying co-articulated sound units. The long-term pitch contour of the vocal system in a time window, combined with the phonemic units utilized to generate the contour, could say something about the particular accent. Our aim is to have a mapping between a particular prosodic behaviour, and the underlying syllable structures. We do not need to link the syllable structure with the prosodic behaviour into one feature vector, as this will be an explicitly modelled via our classification system described later on.

$$\begin{aligned} A_t &= \frac{|A_r| - |A_f|}{|A_r| + |A_f|} \\ D_t &= \frac{|D_r| - |D_f|}{|D_r| + |D_f|} \\ \Delta F0 &= F0_p - F0_v \end{aligned} \tag{6.2}$$

The dynamics of pitch contours are represented using tilt parameters, in the way encoded in [207]. We characterize the possible pitch change ($\Delta F0$), rises, falls, or rises followed by falls in segments by using tilt parameters for amplitude (A_t) and duration (D_t) defined in Equation 6.2. A_r and A_f represent the rise and fall in pitch amplitude with respect to $F0_p$, the pitch peak in the contour. D_r and D_f represent the duration for the rise and fall respectively. This concept is summarized in Figure 6.3. There are other techniques for pitch contour parametrization that we unfortunately did not have time to investigate in this thesis, and further experimentation with different techniques (such as Legendre polynomial expansions in [208, 209]) are recommended.

6.3.1 Context-Dependent GMM-UBM Accent Models

The accent models we propose are based on using prosodic feature vectors to provide context, and the short-term feature vectors for a segment to characterize the syllabic inventory for a particular prosodic behaviour under a particular accent group. Given a prosodic feature vector for a speech segment (which encompasses multiple short-term frames of equal length, which we term 'frame sequences'), we find the closest matching centroid in our reference model, for each prosodic vector in an utterance.

The 'frame sequences' of the speech segment are passed on to the construction of a Gaussian Mixture Model (GMM) for the centroid that is associated with the prosodic feature vector. This concept is demonstrated in Figure 6.4. Each accent is therefore modelled with k sub-GMMs, where the value of k is equal to the number of centroids in the prosodic behaviour space model.

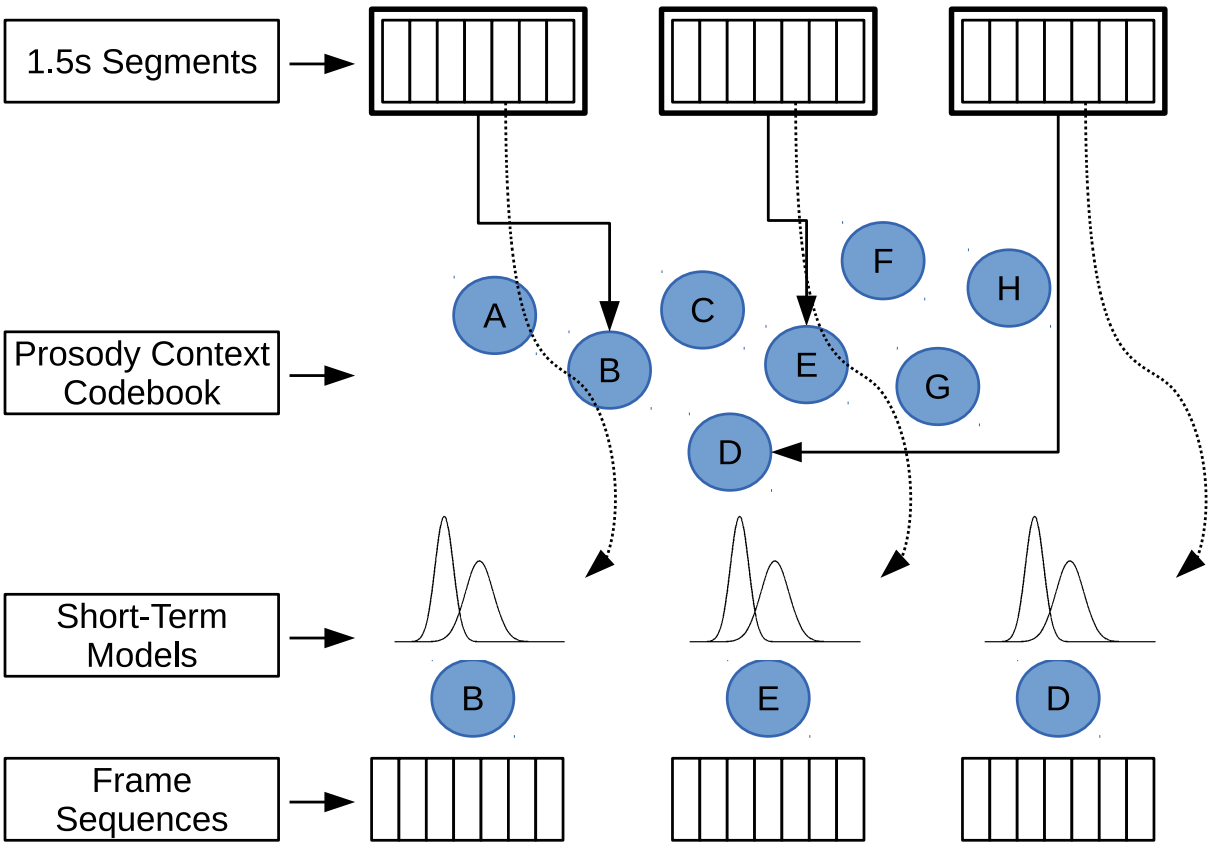


Figure 6.4: An utterance has three segments. Each of the prosodic vectors derived from the segments is associated with one of the centroids of the prosodic space. In this example, segment 1 is associated with centroid B, segment 2 is associated with centroid E, whilst segment 3 is associated with centroid D. For this reason, the short-term vectors from the first segment are used as part of the training set for the short-term accent GMM of prosody index B, the second segment frames are used as part of the training set for the short-term accent GMM of prosody index E, and the third segment frames are used as part of the training set for the short-term accent GMM of prosody index D. Moreover, the short term frames from the each segment, will therefore not be involved in the training of accent GMM of other prosodic regions: frames used to train a model for B are not used for training in E and D etc.

6.3.2 Classification

Given these accent models, the ranking of an utterance U with segments $1 \ldots S$ for a specific accent model a, can be summarized as the sum over the probabilities of each segment, as in Equation 6.3.

$$\mathfrak{R}_a(U) = \sum_{s=1}^{S} \xi(s) \tag{6.3}$$

The term $\xi(s)$ is the rank of the prosodic segment s for the particular accent, and is defined in Equation 6.4, where $n = 1 \ldots N$ is the sequence of N feature vectors for a particular prosodic

segment (the frame sequence for the segment).

$$\xi(s) = \sum_{n=1}^{N} p(s|n) \tag{6.4}$$

Each term $p(s|n)$ is equivalent to $p(s|\lambda)$ such that λ is the accent GMM chosen for a particular segment, as defined in Equation 6.5, by choosing the closest centroid from $1 \ldots K$ for a given frame. The index k of the chosen centroid is then used to obtain a classification of the kth GMM for the accent in question, out of K GMMs constructed during training. Therefore, the short-term vectors for each segment are used to find the closest centroid from $1 \ldots K$ for each frame, resulting in N indexes, one per frame. The index of each targeted centroid is used to calculate the terms $p(s|n) \ldots p(s|N)$.

$$\lambda = \min_{1<k\leq K} d(f_n, \lambda_k) \tag{6.5}$$

Each classifier will give a ranked probability for each segment. The higher the ranked probability, the more likely that segment is generated by the accent.

$$A_c = \max_{1<a\leq A} \mathfrak{R}_a(U) \tag{6.6}$$

The utterance U is given a rank by all accent models, and the accent model with the highest ranking is the final classification A_c of the utterance, as shown in Equation 6.6. Within this scheme, the traditional GMM-UBM scoring (as in Approach I) is a special case of Approach III, where there is only one prosodic context (the whole set), and therefore only one GMM per accent.

6.3.3 Results

The results obtained for Approach III are shown in Figure 6.5. A number of trials were performed for different codebook sizes of the prosodic context. The results show how AID performance deteriorates as the codebook size (and therefore number of GMM models per accent) increases. Similar to the trends observed for Approach I and II, performance generally improves as the number of components used for GMM models increases.

The results in Figure 6.6 compared the best of Approach III, with the previous results for Approach I and II. The results show that the general case of having only one GMM per accent (Approach I) performs slightly worse than Approach III. However, the solution provided

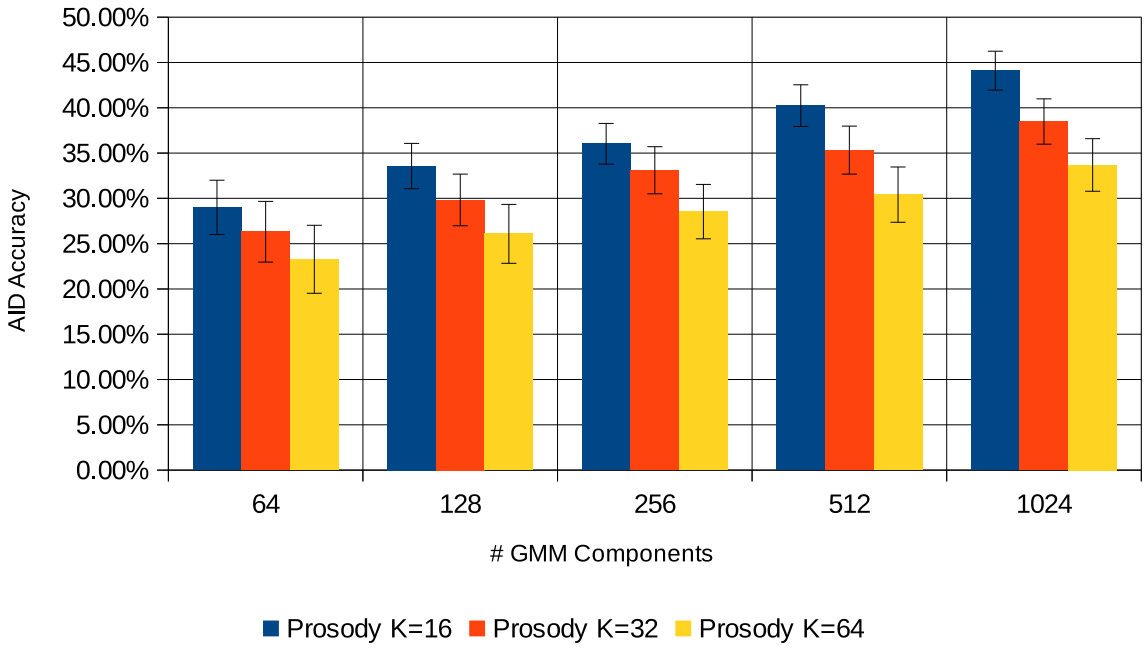


Figure 6.5: Accent identification results for Approach III: Prosodic context-based GMM-UBM classification on short-term feature vectors that include only spectral information.

by Approach II (combined feature vectors of short-term spectral information with prosodic information) gives the best performance.

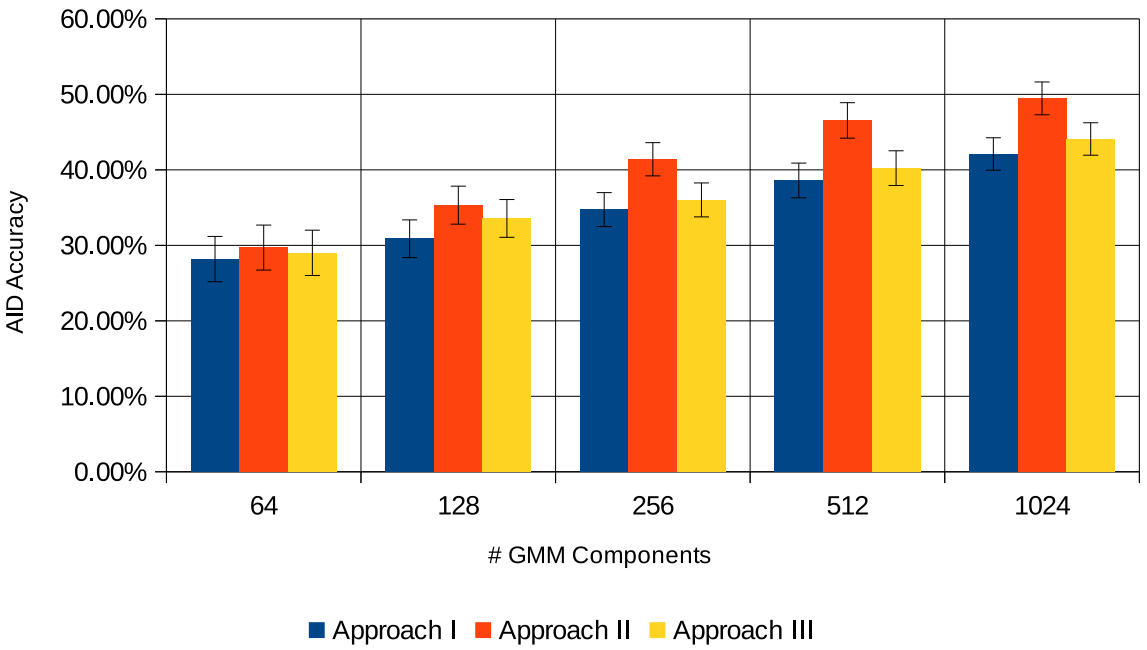


Figure 6.6: Comparison of different GMM-UBM based approaches to accent identification.

6.4 GMM-SVM Class Supervectors (Approach IV)

The GMM-SVM class supervectors approach is a direct extension of Approach I. The same accent-specific GMMs are constructed by MAP-adapating a UBM through data from specific accents. The supervector is extracted as a concatenation of the GMM mean vectors over all the GMM components. Since the GMM is based on training data from multiple utterances and multiple speakers, the resulting supervector is an utterance independent, speaker-independent, but accent-dependent supervector. Each accent is therefore represented by a single supervector.

6.4.1 Feature Extraction

Feature extraction is exactly the same as performed in Section 6.1.1. The dimensionality of the supervector depends on the number of GMM components and the dimensionality of the original frame feature vectors. For a GMM of k components, the final supervector dimensionality is therefore $k \times 62$ dimensions. A sample plot of accent supervectors plotted in low dimension is shown in Figure 6.7.

6.4.2 Classification

For this approach, we employ a number of SVM classifiers from the LIBSVM library. The resulting SVM is then employed to classify individual test utterances. The data from each utterance is used to MAP-adapt the UBM to an utterance-dependent GMM, and the resulting supervector from the GMM is passed to the SVM for classification.

For reference, we document the default parameters for the SVM library at the time of writing, since these may change over time:

- Degree: degree in kernel function, default set to 3
- Gamma: gamma in kernel function, default set to 1/number of features
- Cost: cost parameter, default set to 1
- Shrinking: whether to use shrinking heuristics, default set to 1
- Epsilon: set tolerance of termination criterion, default set to 0.001
- Weight: set the weight parameters of each class, default set to 1

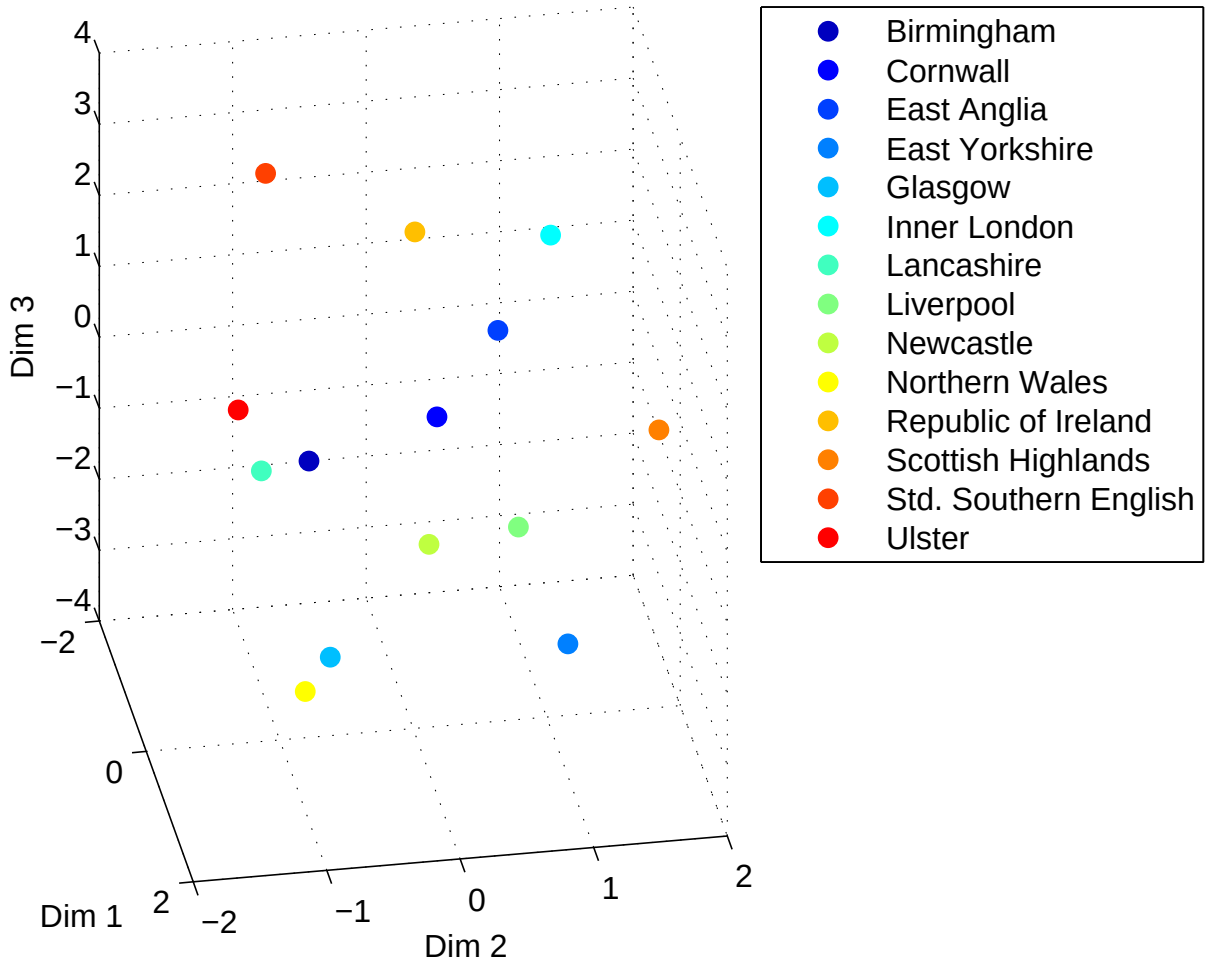


Figure 6.7: Accent supervectors plotted in low dimension obtained by plotting the first three LDA dimensions.

In some of the results we report for SVM classification in this chapter, performance is poor. It cannot be ruled out that these poor results are sometimes obtained due to the default parameter settings reported above. In fact, a search for optimal parameters against a development dataset was not performed. The reason for this was that a development dataset would severely reduce the amount of actual training data available, and would impact performance in the long term, especially for the classifiers we build in later chapters, based on i-Vectors. For this reason, a thorough search for optimal parameters was bypassed, and default configurations used. We therefore do not exclude the possibility that these results could be much improved, if an appropriate data set and parameter finding exercise is performed.

6.4.3 Results

The first experiment is performed on a SVM with a Radial Basis Function (RBF) kernel, and default parameters from the LIBSVM library. The results for this classifier are shown in Figure 6.8.

The results show how the GMM-SVM technique performs worse than the GMM-UBM technique in Approach I. Also, the trend of improved results with the increase of GMM components is not observed with GMM-SVM classification on an RBF kernel. In fact, performance deteriorates rapidly with larger GMM sizes.

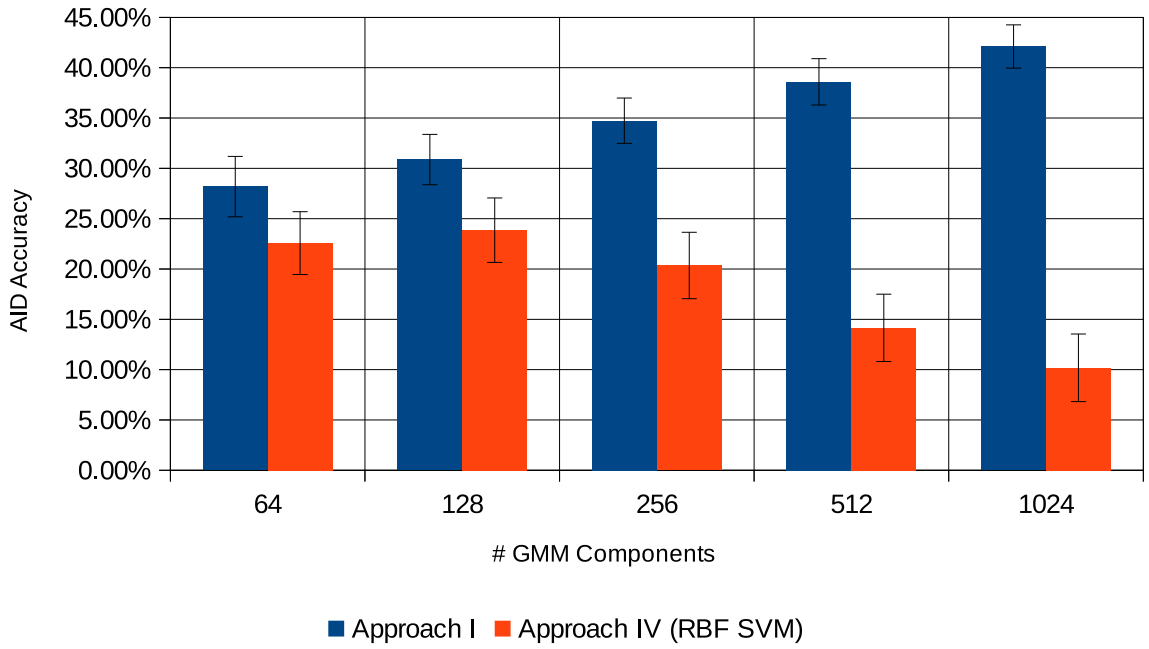


Figure 6.8: Accent identification results for Approach IV with an RBF kernel SVM: GMM-UBM classification (Approach I) performs better in all tests.

The second experiment is performed on a SVM with a linear kernel. The results for this classifier are shown in Figure 6.9. Even in this case, the results for GMM-SVM classification are much worse than those in Approach I, and are interestingly equivalent to an RBF kernel SVM classifier.

The third experiment is performed on a SVM with a polynomial kernel of varying degrees. The results for this classifier are shown in Figure 6.10. As with other SVM classifiers, the performance is poor compared to those in Approach I. Also, performance degrades as the degree of the polynomial kernel is increased.

6.5 GMM-SVM Utterance Supervectors (Approach V)

The previous section focused on SVM classification based on single accent supervectors. In all the experiments, the performance obtained for utterance-based supervector classification was poor. There are a number of possible reasons for this. The supervectors are probably too much

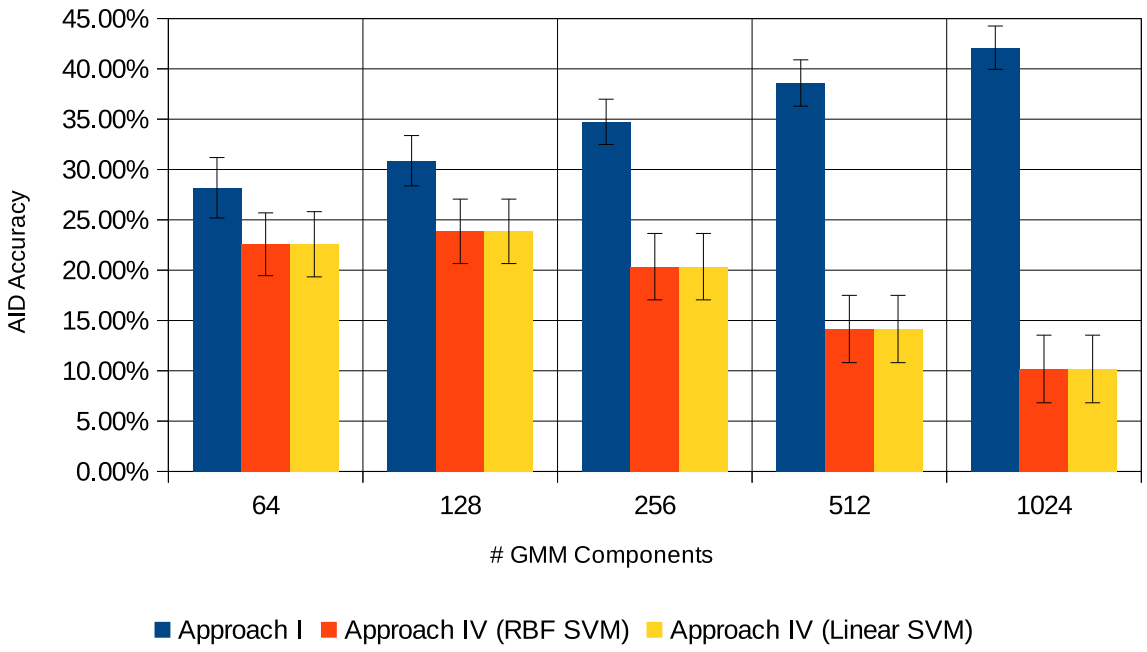


Figure 6.9: Accent identification results for Approach IV with a linear kernel SVM: GMM-UBM classification (Approach I) performs better in all tests. Interestingly, the performance for an RBF and a linear kernel is equivalent.

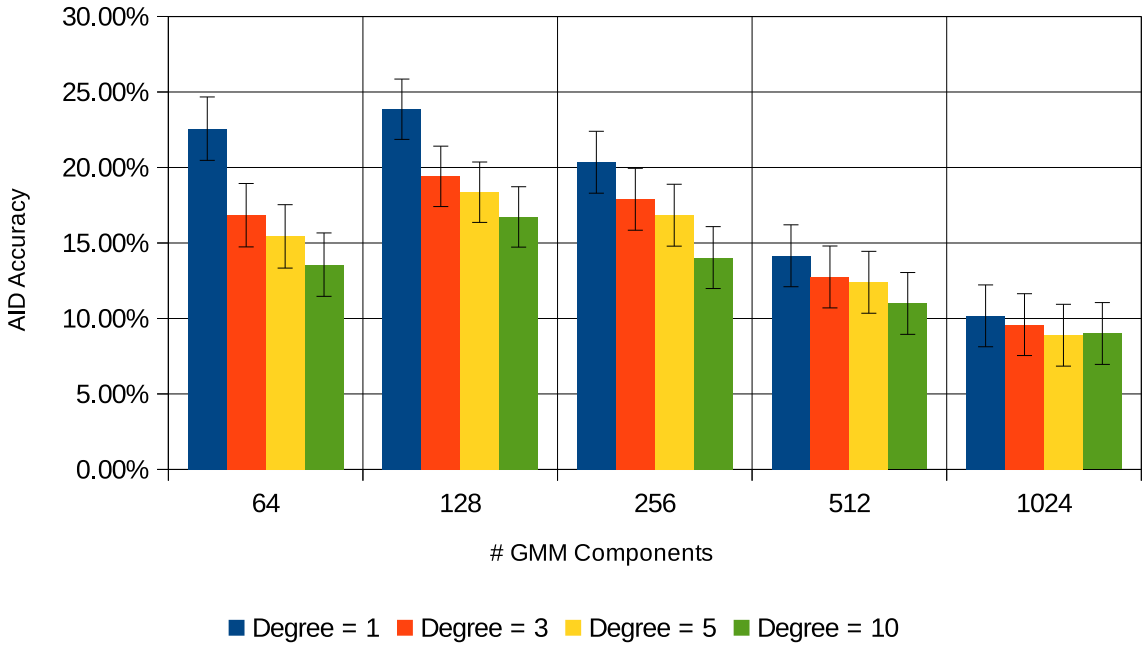


Figure 6.10: Accent identification results for Approach IV with a polynomial kernel SVM: GMM-UBM classification (Approach I) performs better in all tests. The performance decreases rapidly for kernels of polynomial degree >1.

influenced by speaker information. The supervector itself does not represent just the accent, but also all the other information contained in the utterances used to train the accent supervector. This presents a problem for an SVM classifier — the distinction between accent classes cannot

be appropriately modelled, and classification performance suffers. It is curious that Approach I, based on GMM-UBM classification, performs better than all the attempts in Approach IV (techniques that use SVMs). After all, the supervector is simply a vectorized representation of the GMM for each accent. We postulate that the SVM was used in a sub-optimal manner, which could explain the performance difference. The SVM implementation is based on default configurations as provided by the LIBSVM library.

In this section, we try out a different form of GMM-SVM classification. The goal this time is to model all utterances as individual supervectors. Therefore instead of having a single supervector to represent an accent, each accent is represented by a number of supervectors, one per utterance in the training set. Hence, as far as front-end feature extraction is concerned, the configuration is exactly the same as in Approach I and Approach IV. However, the frames of each utterance are used to MAP-adapt the UBM into a GMM, and consequently a supervector.

6.5.1 Results

The first experiment is performed on a SVM with a Radial Basis Function (RBF) kernel, and default parameters from the LIBSVM library. The results for this classifier are shown in Figure 6.11. The results show how the GMM-SVM technique for utterance based supervectors now performs better than both Approach I and Approach IV for the same classifier. There is no improvement resulting from models created by supervectors from GMMs of a different number of components, and results are close across the board.

In these results, we can not guarantee that the SVM parameters have been optimally set, and it is probably that the default configurations causes increasingly bad performance as the supervector dimensionality increases. In the next experiment, PCA is performed on the supervector training set (and consequently the PCA mapping is applied during testing time as well), to see if reducing the dimensionality of the supervectors has any effect on performance under the same RBF kernel SVM and same default LIBSVM configuration. The results, for various PCA dimensionality is shown in Figure 6.12.

The performance obtained when classifying supervectors that have been reduced to a low dimensionality by PCA degrades very slightly compared to the performance obtained without dimensionality reduction. The dimensionality reduction however allows us to utilize the PCA-reduced supervectors to create another feature extraction layer prior to classification, one that could compensate for the differences in supervectors of an accent caused by utterance

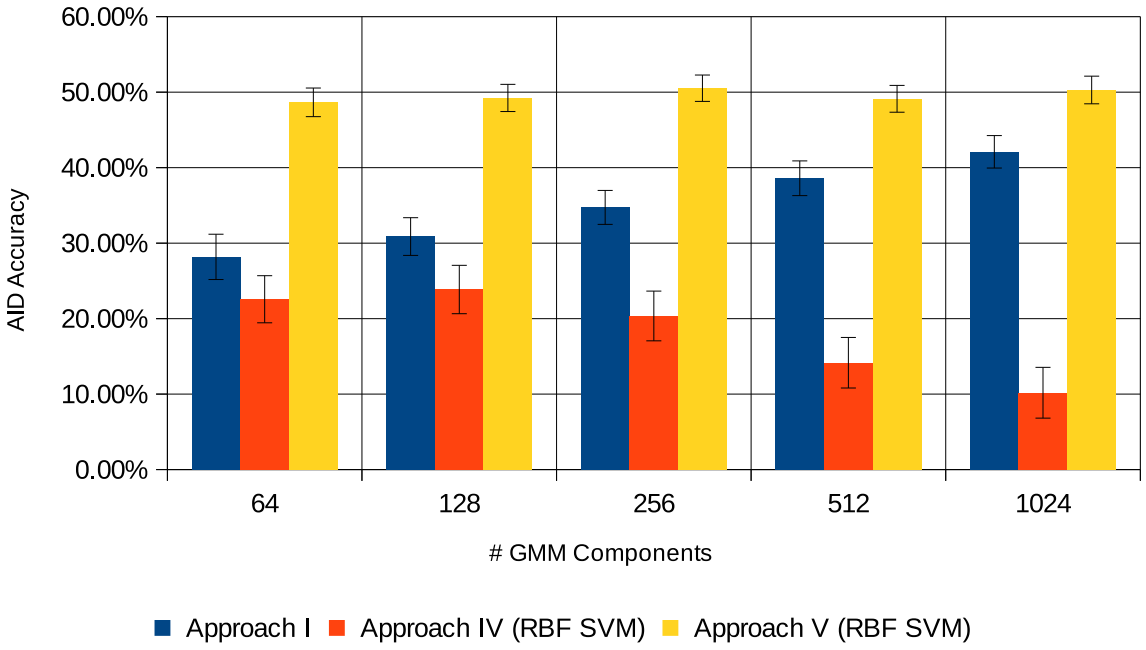


Figure 6.11: Accent identification results for Approach V with an RBF kernel SVM: This new approach performs better than GMM-UBM classification (Approach I) and GMM-SVM with RBF kernel for single supervectors per accent (Approach IV).

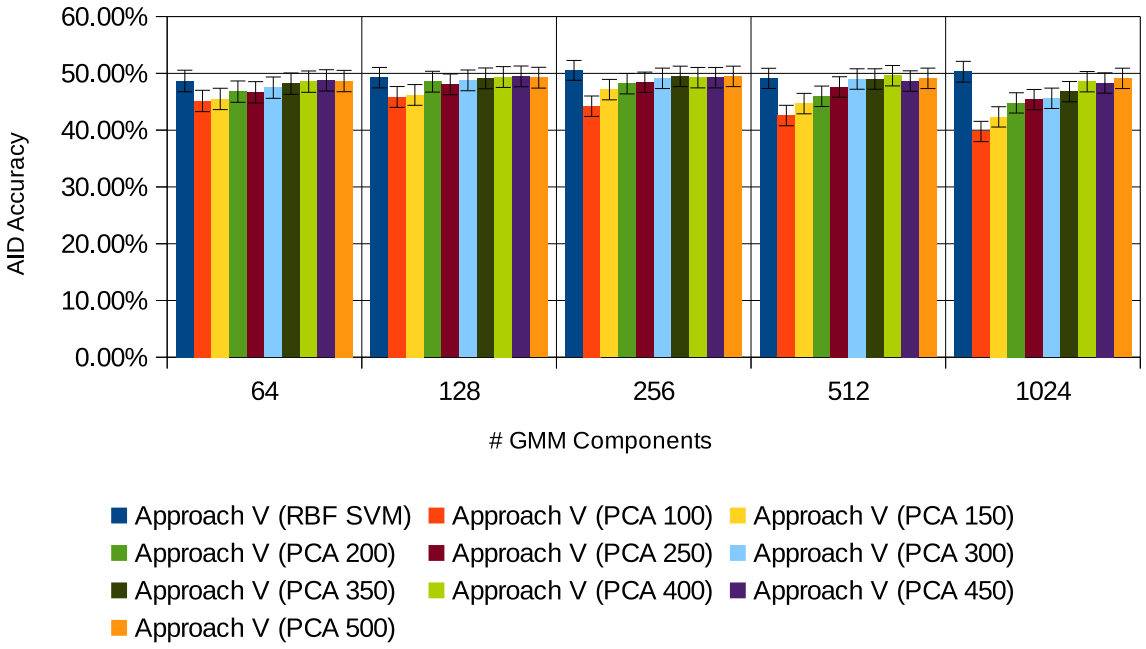


Figure 6.12: Accent identification results for Approach V with an RBF kernel SVM and PCA applied to supervectors: Though the results without PCA are the best, PCA does not really degrade performance very much at a dimensionality of 250 to 500.

and speaker-specific variations. In the following plots, we can see a set of supervectors after PCA reduction is performed (Figure 6.13), and consequently, after LDA projection is performed (Figure 6.14).

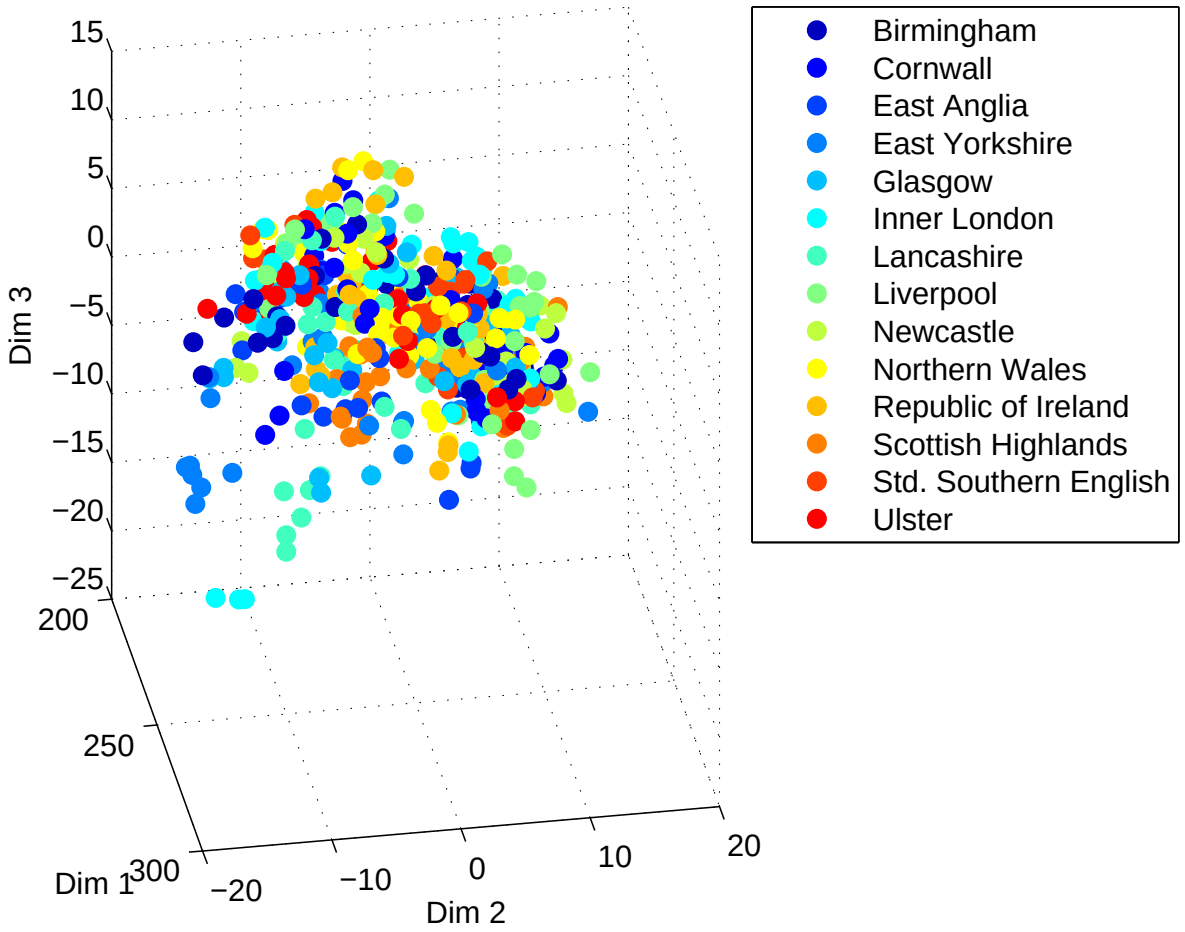


Figure 6.13: Dimensionality reduced supervectors after PCA is applied to GMM-UBM supervectors. (PCA dimensionality = 100)

When both PCA and LDA are applied after each other, the accent clusters are much more discernible than with just the application of PCA. Also, the class separation is potentially better when the dimensionality of PCA is increased from 100 to 500. The plots in Figure 6.16 and Figure 6.17 are the same supervectors, being mapped by PCA and LDA, with a higher PCA dimensionality of 500. The higher PCA dimensionality results in much more discernible accent clusters once LDA is applied.

The next test to perform is therefore SVM classification via a default RBF kernel over PCA and LDA-reduced supervectors. The results for this experiment are shown in Figure 6.15. The best results are observed on supervectors that have been PCA-reduced with high dimensionality, and contrary to Approach I, on low order GMM sizes. Finally, a comparison of results of different approaches is summarized in Figure 6.18. The best results are obtained by Approach V. The LDA stage of supervised training is strongly conditioning the supervectors to attenuate non-class specific information, and the plots showing clearer accent clusters confirm this. Surprisingly, changing SVM kernel and parameters does nothing to aid the AID classification performance

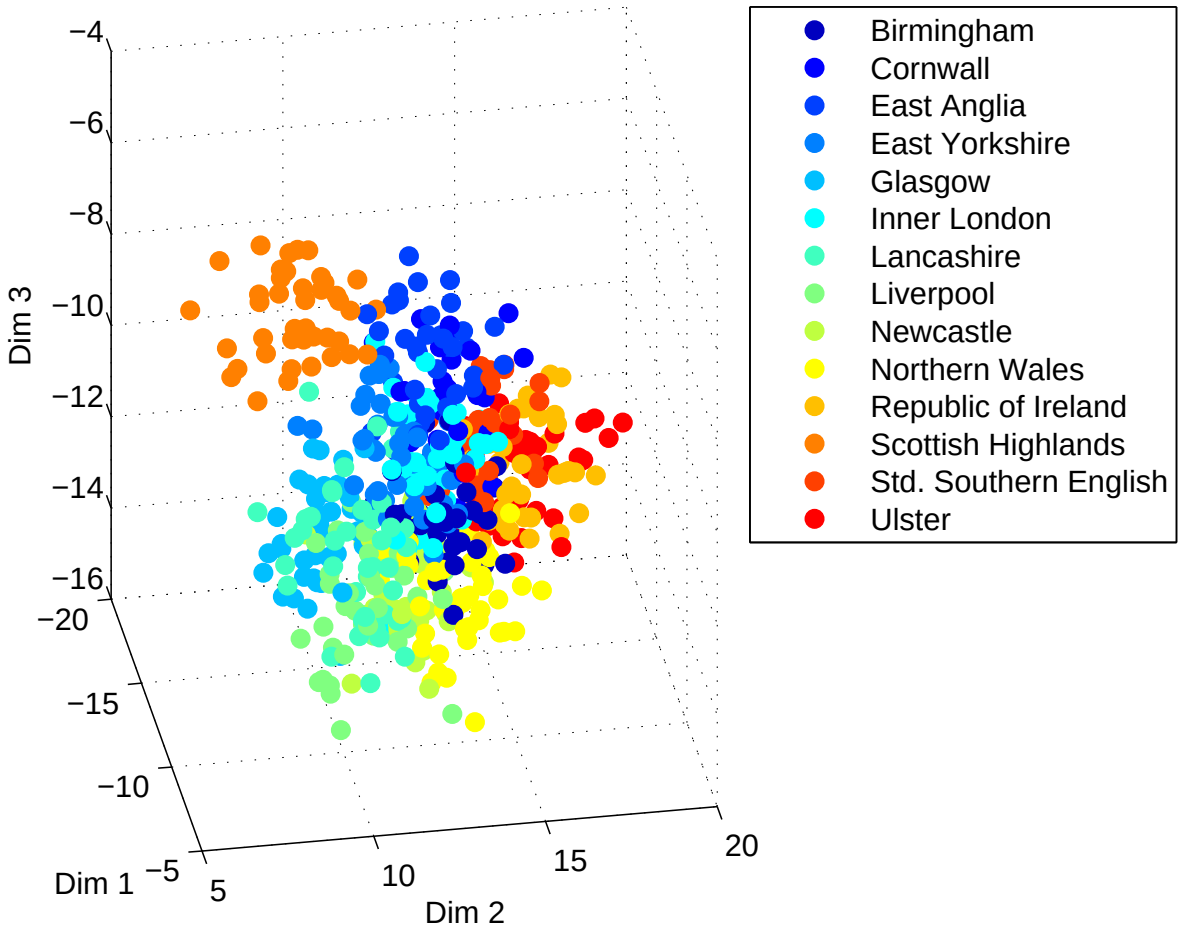


Figure 6.14: Dimensionality reduced supervectors after PCA and LDA are applied to GMM-UBM supervectors. (PCA dimensionality = 100)

prior to the PCA+LDA stages, except give slighlty better performance than GMM-UBM when multiple supervectors per accent are used for training.

6.6 Accent Confusion Analysis

As reported in the previous section, the best accent identification performance obtained so far using standard classification methods that are popular across gender/speaker/language identification is given by Approach V: utterance-specific supervectors, reduced in dimensionality by PCA, and further optimized by LDA and combined with a SVM classifier with a RBF kernel. It is interesting to look at which accents are being confused with which other accents. This analysis can provide some insight into whether some accents are harder or easier to classify, and in the case of wrongly classified accents, whether the chosen wrong classification has some basis in acoustics and phonetics, rather than being purely a consequence of the classifiers used. A confusion matrix showing the correct versus predicted classification of utterances of the corpus

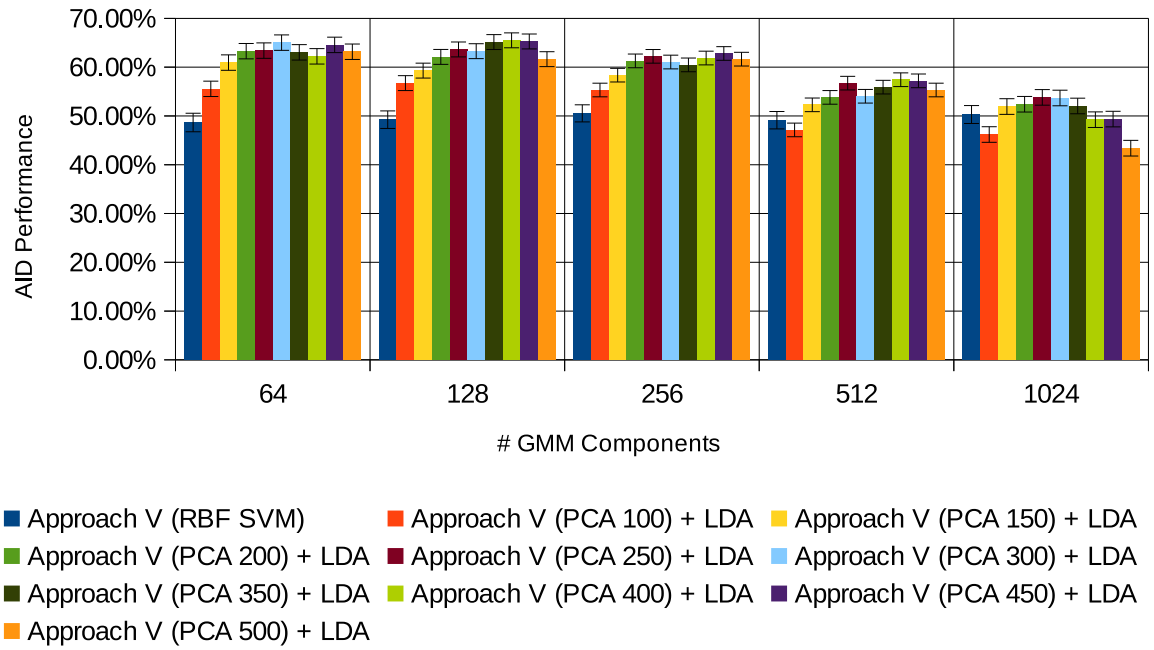


Figure 6.15: Accent identification results for Approach V with an RBF kernel SVM and PCA+LDA applied to supervectors: These results show that LDA produces more discernable clusters which aid the SVM classifier, giving some relatively good AID classification performance, especially on the low order GMM sizes and high PCA dimensionality.

for Approach V is given in Table 6.3. A list of closest confusions per accent is given in Table 6.1. There does not seem to be any direct relation to the broad accent geographic formation (see Section 4.2) in most cases based on the confusions across accents.

Another analysis which can be looked at is the proximity of accents to each other after the training phase is completed, through the Euclidean distance of the mean of each accent cluster. This is shown in Table 6.2. Again, there is no regular pattern of accents clustering close to each other in terms of the broad accents of the British Isles as in Section 4.2. This is overall a bit disappointing, in that the learnt structure of accents from acoustic, rather than acoustic-phonetic feature vectors does not define accents in the same terms as one would expect accents to morph and overlap across geographic regions. It implies, however, that there is some element of accent information that is defined purely acoustically/intonationally. This will be discussed even further in later chapters as we approach the problem with different techniques.

6.7 Summary

In this chapter we have evaluated a number of standard classifiers classically employed for speaker and language identification. We have also tried to utilize some additional structures

Table 6.1: Ordered list of closest confusions per each accent as given by the Approach V classifier. Where no confusions are made, columns are left empty.

Accent	Closest Accents												
brm	nwa	ean	ilo	eyk	crn	lvp	ncl	sse					
crn	ilo	ean	nwa	sse	brm	lvp	ncl	eyk	shl	uls			
ean	brm	sse	crn	ilo	lan								
eyk	ilo	lan	shl	ean	brm	sse	ncl	nwa	roi				
gla	ncl	ilo	brm	shl	ean	eyk	lvp	nwa					
ilo	brm	ean	eyk	gla	lan	crn	nwa	roi					
lan	eyk	nwa	ilo	brm	ean	ncl	shl						
lvp	nwa	brm	eyk	lan	crn	ean	gla	ilo	ncl				
ncl	nwa	brm	ilo	lvp	ean	roi	sse	crn	eyk	gla			
nwa	brm	crn	lvp	roi	ilo	lan	ncl	eyk	sse				
roi	uls	nwa	brm	crn	ilo	lvp	lan						
shl	lan	ncl											
sse	brm	ean	nwa	crn	ilo	ncl	eyk	lan	roi				
uls	roi	crn	brm	ilo	lan	sse							

Table 6.2: Ordered list of closest accents for every other accent as given by the Approach V training results.

Accent	Closest Accents												
brm	ean	eyk	ilo	nwa	gla	ncl	lvp	lan	sse	crn	uls	roi	shl
crn	ilo	nwa	ean	gla	ncl	sse	uls	brm	lan	eyk	lvp	roi	shl
ean	ilo	brm	ncl	crn	lan	gla	eyk	sse	nwa	lvp	uls	roi	shl
eyk	brm	lan	sse	lvp	ean	ilo	ncl	gla	nwa	uls	roi	crn	shl
gla	ncl	ilo	ean	nwa	brm	crn	eyk	lan	lvp	sse	uls	shl	roi
ilo	ean	crn	ncl	gla	nwa	brm	eyk	lan	sse	lvp	uls	roi	shl
lan	eyk	ean	ncl	ilo	brm	lvp	gla	sse	nwa	uls	shl	roi	crn
lvp	ncl	eyk	brm	ilo	lan	ean	nwa	gla	sse	crn	shl	uls	roi
ncl	nwa	gla	ilo	ean	lvp	lan	brm	eyk	crn	sse	uls	shl	roi
nwa	ncl	crn	ilo	brm	ean	gla	sse	eyk	lvp	lan	uls	roi	shl
roi	uls	sse	eyk	lan	nwa	ilo	ean	shl	crn	brm	gla	ncl	lvp
shl	lan	uls	sse	gla	ilo	eyk	roi	ncl	ean	lvp	crn	brm	nwa
sse	eyk	ean	ilo	brm	nwa	crn	uls	lan	roi	gla	ncl	lvp	shl
uls	roi	sse	ean	ilo	lan	crn	ncl	shl	gla	eyk	brm	nwa	lvp

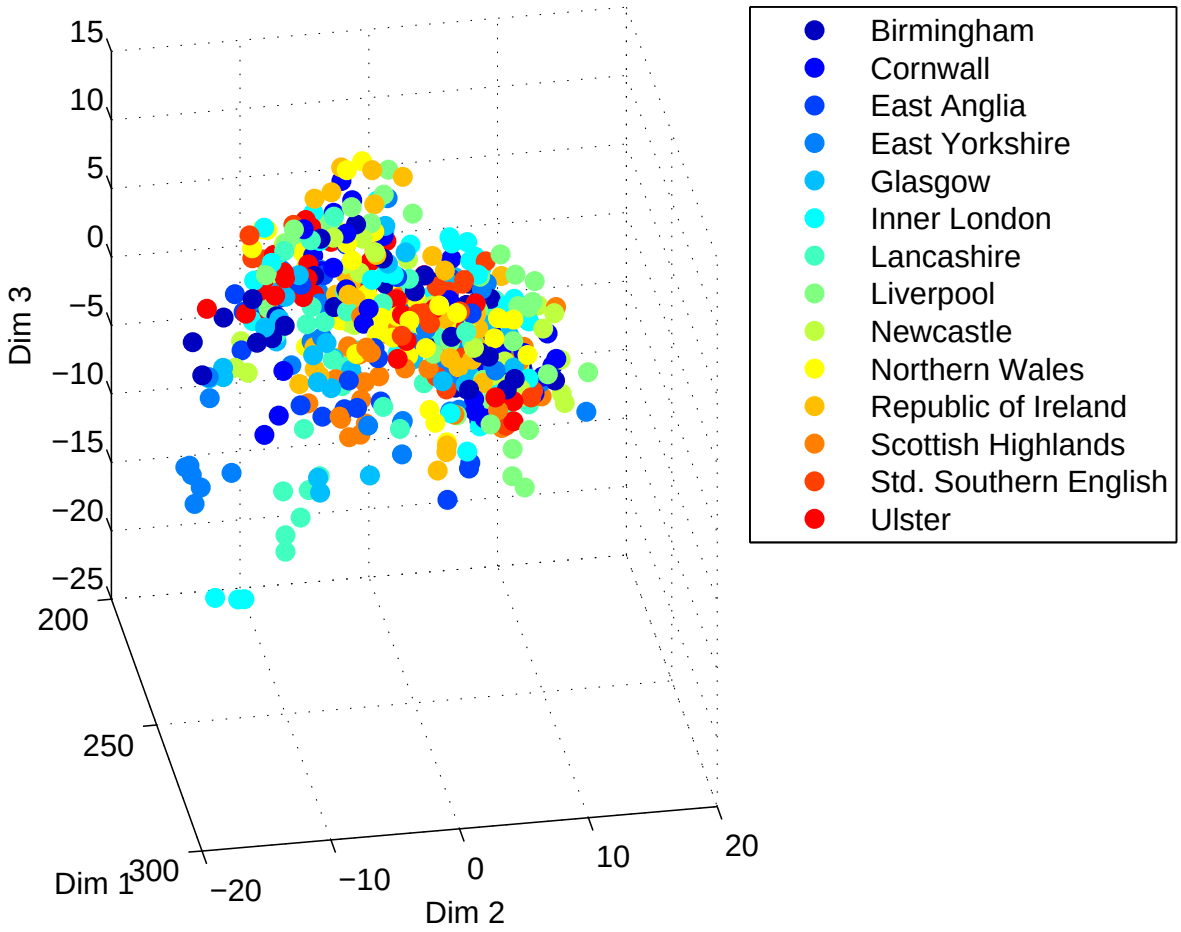


Figure 6.16: Dimensionality reduced supervectors after PCA is applied to GMM-UBM supervectors. (PCA dimensionality = 500)

to give prosodic meaning and context to speech frames within the classifier — with no real gain in performance. The best gain obtained was based on first reducing the dimensionality of utterance-dependent supervectors by PCA, and then finding a linear discriminant projection for the classes via LDA, which of course has some variability compensating power over the supervectors, and performance reaching around 65% accuracy. This is a good first step. However, the error margin is still large. In the next chapters, we set out the first applications of i-Vector modelling to the problem of accent identification, and a number of fusion enhancements for speaker compensation.

The experiments in this chapter, especially the ones that give reasonable results on supervectors, are based on long utterances of around 30s of speech (once silence and unvoiced frames are removed). The nature of MAP-adaptation of a UBM suggests that performance would drop considerably if the duration of utterances is much shorter than this. This is a general problem in speech classification problems, including speaker and language identification. The focus of this thesis in the next chapters will be to build even better classifiers on the same 30s utterances.

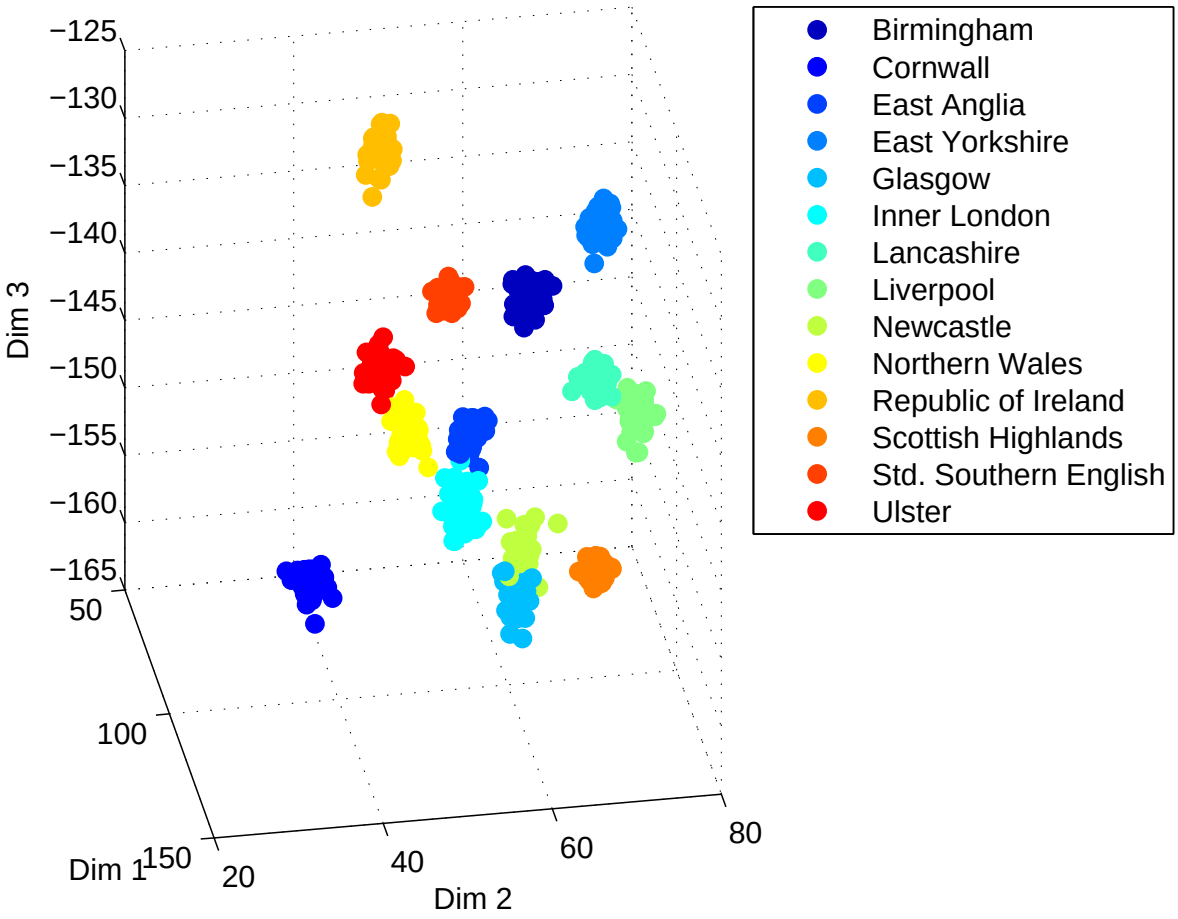


Figure 6.17: Dimensionality reduced supervectors after PCA and LDA are applied to GMM-UBM supervectors. (PCA dimensionality = 500)

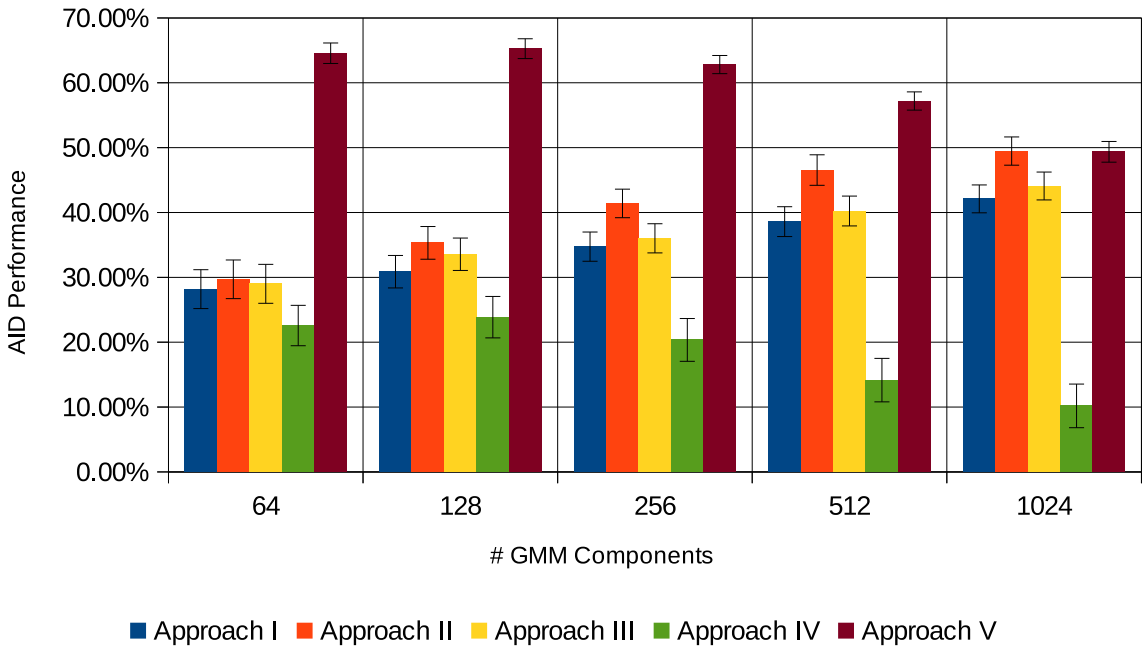


Figure 6.18: A comparison of the best set of results from all approaches in this chapter.

Once we obtain a highly-optimised AID system, we will then look into how such a system scales with shorter utterances, since good performance on a few seconds of data is very desirable in these systems.

Another aspect of our analysis in future chapters will be a direct comparison of the confusion matrix obtained by different classifiers. Will other (and better) classifiers simply enhance the confusion matrix we obtained in this chapter? Or will the "proximity" of accents change according to the different accent patterns learnt by the different classification systems? Moreover, will there by any classifier that gives some semblance of geographic/linguistic accent proximity? So far, the approaches discussed in this chapter give no such insight into accents, and we find this to be disappointing if we compare the results with methods that tackle accent classification from an acoustic-phonetic perspective such as the ACCDIST metric and phonotactic systems as the ones described in Chapter 3.

Table 6.3: Confusion matrix (in %) of correct vs predicted classification of utterance for the 14 accents of the British Isles. Average accent recognition accuracy is of 65%. The diagonal, which represents the correct (no confusion) results is in **bold**, whilst off-diagonals (confusion) equal or greater than 8% are marked in **red**.

	brm	crn	ean	eyk	gla	ilo	lan	lvp	ncl	nwa	roi	shl	sse	uls
brm	**61.67**	1.67	**10.00**	3.33	0.00	5.00	0.00	1.67	1.67	**13.33**	0.00	0.00	1.67	0.00
crn	6.67	**41.67**	**8.33**	1.67	0.00	**11.67**	0.00	5.00	5.00	**8.33**	0.00	1.67	**8.33**	1.67
ean	**14.04**	3.51	**66.67**	0.00	0.00	3.51	3.51	0.00	0.00	0.00	0.00	0.00	**8.77**	0.00
eyk	4.00	0.00	5.33	**58.67**	0.00	**9.33**	**9.33**	0.00	1.33	1.33	1.33	6.67	2.67	0.00
gla	3.33	0.00	1.67	1.67	**73.33**	5.00	0.00	1.67	**8.33**	1.67	0.00	3.33	0.00	0.00
ilo	**9.52**	3.17	**9.52**	6.35	4.76	**57.14**	4.76	0.00	0.00	3.17	1.59	0.00	0.00	0.00
lan	3.17	0.00	1.59	**26.98**	0.00	4.76	**53.97**	0.00	1.59	6.35	0.00	1.59	0.00	0.00
lvp	5.00	1.67	1.67	5.00	1.67	1.67	5.00	**65.00**	1.67	**11.67**	0.00	0.00	0.00	0.00
ncl	5.00	1.67	3.33	1.67	1.67	5.00	0.00	5.00	**56.67**	**13.33**	3.33	0.00	3.33	0.00
nwa	**11.11**	4.76	0.00	1.59	0.00	3.17	3.17	4.76	3.17	**61.90**	4.76	0.00	1.59	0.00
roi	3.33	3.33	0.00	0.00	0.00	3.33	1.67	3.33	0.00	5.00	**71.67**	0.00	0.00	**8.33**
shl	0.00	0.00	0.00	0.00	0.00	0.00	3.03	0.00	1.52	0.00	0.00	**95.45**	0.00	0.00
sse	**16.67**	6.25	**10.42**	4.17	0.00	6.25	2.08	0.00	6.25	**8.33**	2.08	0.00	**37.50**	0.00
uls	1.67	3.33	0.00	0.00	0.00	1.67	1.67	0.00	0.00	0.00	**13.33**	0.00	1.67	**76.67**

Chapter 7

Accent Identification in i-Vector Space

So far we have looked at the problem of AID from the perspective of the most popular speech classification algorithms prior to the development of the JFA and i-Vector paradigms. We have shown that AID is subject to quite large errors when using standard techniques such as the GMM-UBM and GMM-SVM methods, which, on clean recorded data, usually report very good performance on problems such as speaker and language identification. It is highly likely that some of the fine-grained accent distinctions of the British Isles are blurred or obscured by individual speaker differences. The phonetic differences, which are not being modelled here explicitly as is done in say, a phonotactic system, could be very easily obscured when compared to a similar problem in structure such as language identification, where many phonetic differences are observed between one language and another. We can recall the LID task is apparently not too hard even for tamarin monkeys. So an obvious approach is that if it is to cancel out all, or most, of the speaker variability within a class. We have attempted this in the GMM-SVM system already through the use of LDA, and the results obtained showed some promise.

This chapter will focus on the application of the i-Vector paradigm to the problem of accent identification. We will initially assess the effect of the i-Vector paradigm on the AID problem, and compare it to the baseline approaches we discussed in the previous chapter. Furthermore, we will then propose a number of additions to the paradigm, specifically designed to improve results on the AID problem in question — that of native accents of a language. A number of points will be highlighted. Firstly, the trend of the i-Vector paradigm becoming the standard in speech classification problems is also confirmed here, with i-Vector systems giving better results than GMM-UBM and GMM-SVM systems. Secondly, we will highlight a possible

incompleteness resulting from the application of feature dimensionality reduction techniques in i-Vector space, and suggest ways to circumvent this, obtaining even better results. Thirdly, we will show how certain standard projections that have been demonstrated to work well for problems such as speaker and language identification from speech do not necessarily apply that well to the problem of accent identification.

7.1 Frontend and UBM Construction

The i-Vector paradigm is a direct extension of the GMM-UBM method (Approach I). To be able to make direct comparisons, the feature frontend of utterances and the construction of the UBM is exactly the same as in the case for Approach I, which we reproduce below for completeness.

To extract features from an utterance, we first perform voice activity detection to remove silent portions of speech. Following this, 13-dimensional MFCC vectors on the speech utterance, with a window of 30ms and a frame rate of 15ms are extracted over 30 filters spread out to 11025 Hz. Each MFCC vector is converted into a 49-dimensional shifted delta cepstra vector with 7-1-3-7 parametrization. The original MFCC features are then warped to a standard normal distribution with a 3 second time window to minimize effects of channel mismatch, and these warped features are concatenated to the SDCs to form a final set of 62-dimensonal feature vectors.

Training of the universal background model (UBM) is based on the codebook splitting criteria defined in the previous chapter. About eight hours of data is available for training and testing, and therefore, training is performed on approximately five hours of data, rotated for each of the three test sets. If we utilize standard UBM construction techniques of direct estimation, when using a large number of mixture components, there is a high chance of running into problems of components having very small variances (singularities). To circumvent this problem, we use a slower, but more stable way of estimating a UBM from a few hours of training data. The reader is referred to Section 6.1.2 for full details.

7.2 The i-Vector Model

The first uses of total variability and i-Vector methods for speech classification were in the area of speaker verification. The i-Vector representation was based on the success of the Joint Factor

Analysis (JFA) technique. For the purposes of speaker identification, factor analysis is used to construct a low-dimensional subspace, termed the total variability space. This space contains factors of both speaker and channel variability. Unlike JFA, all the variability is contained in a single subspace, whereas each kind of variability is modelled in an explicitly separate subspace in JFA. Once the total variability space is estimated (using the training set only), various methods of intersession compensation can be performed. For the purpose of speaker identification, one would perform compensation say, on channel effects, to retain only speaker-discriminatory information. The premise of a representation of the data in total variability space is that a universal background model (UBM), trained on data from multiple speakers, can be adapted to a given utterance, creating an utterance-dependent Gaussian mixture model (GMM).

$$M = m + Tw \tag{7.1}$$

If we assume that the GMM supervector for a speaker is entirely observable and estimated correctly, then the i-Vector model decomposes the GMM supervector as an additive component to the original UBM, as in Equation 7.1, with M being the GMM supervector, m is the UBM supervector, T is a factor loading matrix for the total variability subspace, and w is the total factor (or i-Vector), a random vector which is a point estimate of an utterance in the total variability subspace with a normal distribution $\mathcal{N}(0, I)$.

The derivation and calculation of the i-Vector is described earlier in this thesis and the reader is referred to Section 3.6.5 for full details. The i-Vector representation technique has been successfully applied to language recognition, and we treat accent recognition as a similar problem, although the differences between classes are much finer in accent recognition. In speaker identification, the total variability space T is estimated using all utterances of a particular speaker as belonging to the different speakers altogether. The same concept is used in language or accent classification, where every utterance is considered as coming from a different language or accent class. This is because each utterance has variability due to both speaker differences and language/accent differences.

7.3 Classification of i-Vectors via LDA (Approach VI)

In the first trial of i-Vector modelling, we configure different i-Vector systems based on the number of UBM components, and the number of factors in the total variability space. The

i-Vectors are then compressed with LDA, and classified with an LDA classifier.

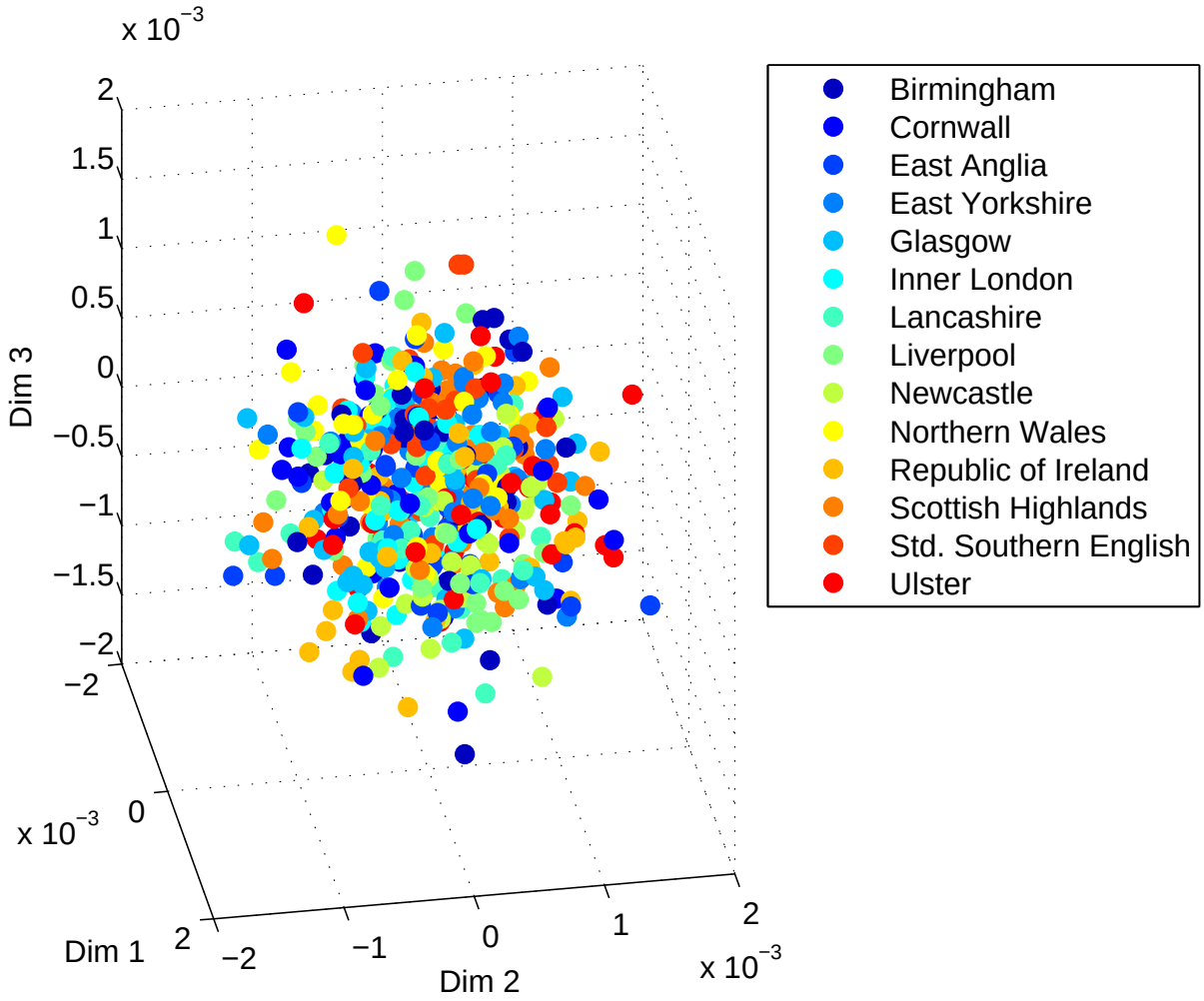


Figure 7.1: Utterances from various accents are transformed as point estimates in the total variability subspace. First three dimensions of the data are shown.

The i-Vectors in the total variability subspace (Figure 7.1) replace the PCA subspace utilised in Approach V. Following the supervised learning procedure of LDA, the i-Vectors are transformed into a lower-dimensionality subspace of 13 dimensions (LDA gives a dimensionality of $n - 1$ where n is the number of classes). This is shown in Figures 7.2 and 7.3. These figures show different degrees of separation between accent classes. Figure 7.2 is based on having a total of 100 factors in the i-Vector subspace, whilst Figure 7.3 is the result of 400 factors in the i-Vector subspace. Both projections, however, are based on the same UBM of 64 components. The larger subspace shows more separation between classes. The question is whether more factors allow for better class separation, or whether this is due to over-fitting factors.

Our testing explores different combinations of UBM component sizes and factor dimensions. The results are shown in Figure 7.4. In this case, classification is also done via LDA. There are a number of things to point out. Firstly, the performance of the i-Vector system gets better as

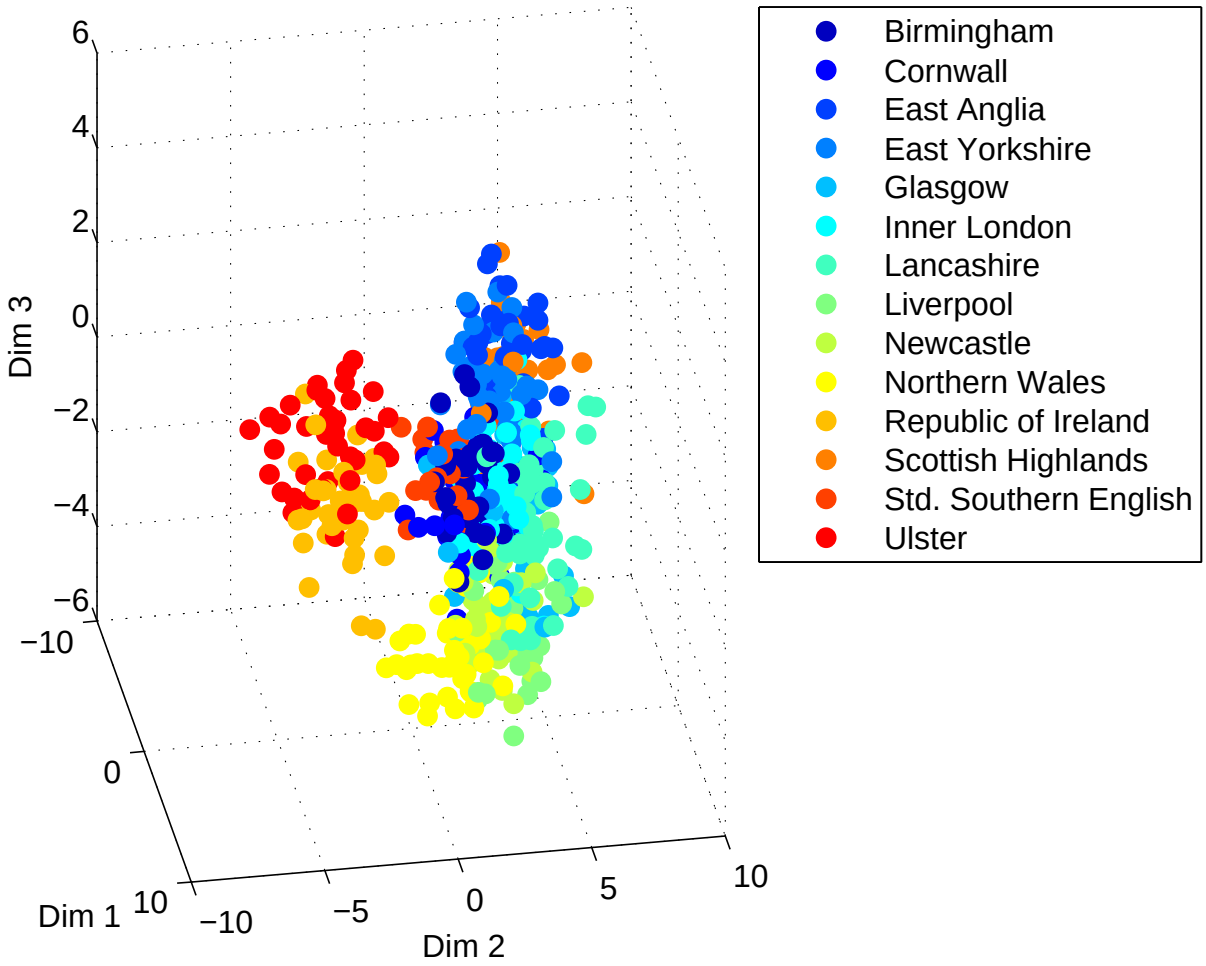


Figure 7.2: Utterances from various accents are transformed as point estimates in the total variability subspace (100 factors), which are then passed on to LDA, resulting in maximally linear discriminant formation between the classes. The first three dimensions of the data obtained by LDA reduction are shown.

more components are used to train the UBM model. However, this performance improvement is not a general trend for every factor dimensionality. For lower factor sizes of 100 and 150, the performance increases as the number of components increases but the same cannot be said for higher factor dimensions in the i-Vector space. Also, even though we observed tighter cluster formations with higher factor dimensions, the results on accent identification are poorer. This suggests that a careful selection of i-Vector model parameters has to be made, as the AID accuracy ranges from 39% to 74%.

7.4 Classification of i-Vectors via QDA (Approach VII)

The LDA classification assumes that there is a linear boundary between all classes. From the low-dimensional plots we have shown earlier, whilst some accents seem to be roughly separable

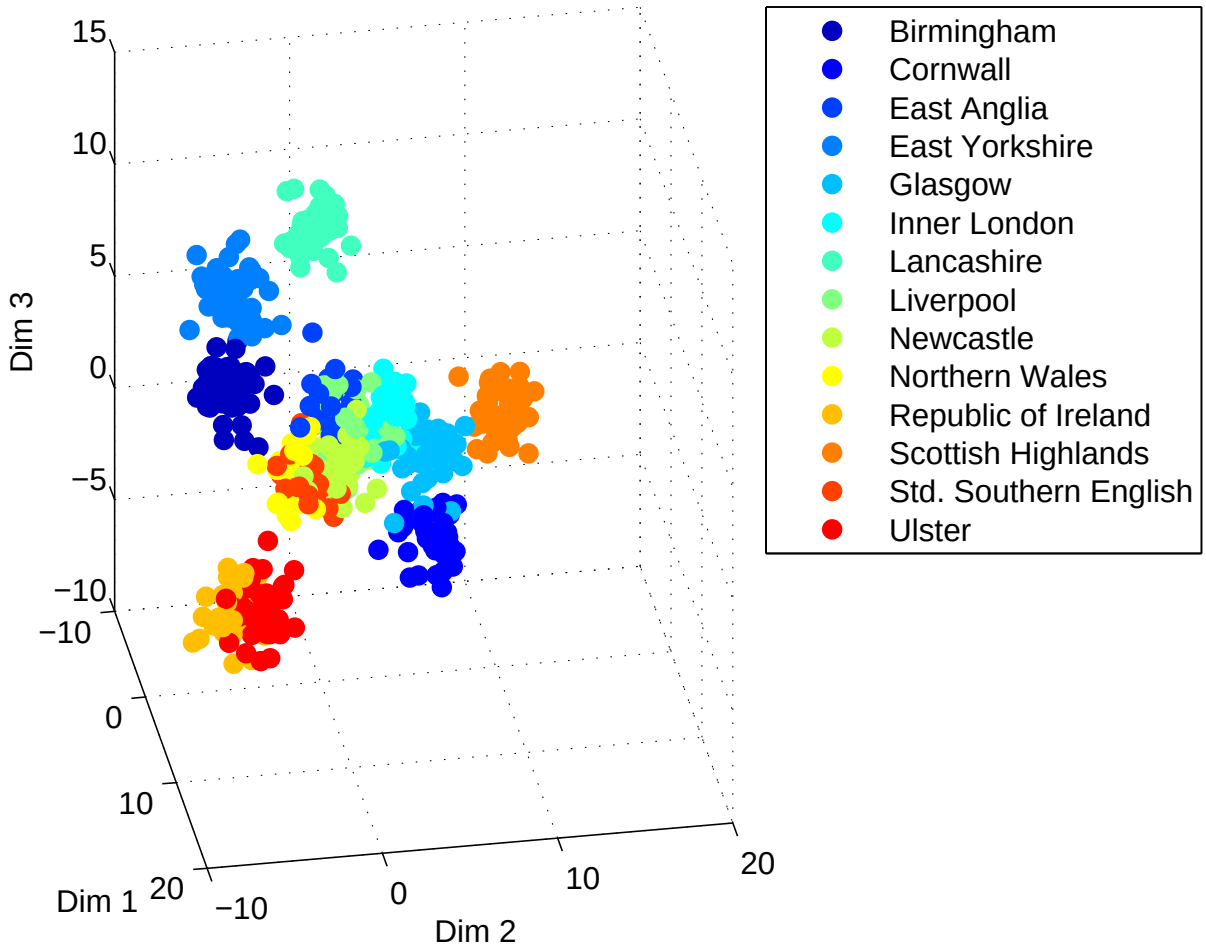


Figure 7.3: Utterances from various accents are transformed as point estimates in the total variability subspace (400 factors), which are then passed on to LDA, resulting in maximally linear discriminant formation between the classes. The first three dimensions of the data obtained by LDA reduction are shown.

linearly, there are several accent clusters that overlap each other to some extent. It is interesting to see whether a more general form of discriminant plane such as that given by Quadratic Discriminant Analysis (QDA) gives any improvement on the classification results. Figures 7.5 and 7.6 show the accent class covariances, and the separating discriminant planes for LDA and QDA respectively.

Here, testing explores different combinations of UBM component sizes and factor dimensions. The results are shown in Figure 7.7. The trend of obtaining better classification results with a larger number of components for the UBM continues in this case as well. Also, lower factor dimensions give better AID accuracy performance than larger factor dimensions. General performance however, is poorer for QDA classification than for LDA classification. The apparently more powerful QDA classification does not solve the problems of large overlap of accent clusters, clearly observable in Figures 7.5 and 7.6. The AID accuracy obtained in this

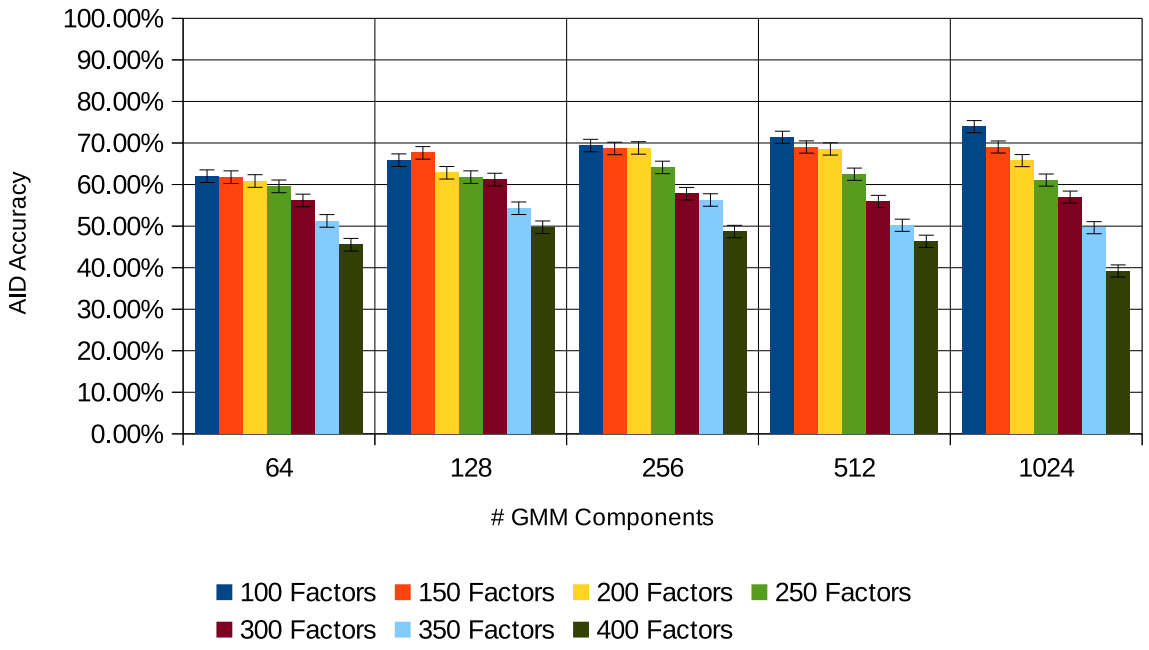


Figure 7.4: The first trial of the i-Vector paradigm on accent identification.

approach ranges from 32% to 65%.

7.5 Classification of i-Vectors via SVMs (Approach VIII)

The previous approach shows that a more powerful classifier may not necessarily give the best results. Linear classification by LDA gives considerably better results than QDA, based on quadratic boundaries between classes. In this approach, we perform trials with linear and RBF kernel SVMs to see if there is any difference or improvement in AID accuracy.

Linear discriminants and linear support vector machines have very common characteristics in that they are both methods that find a linear separation between classes. The added complexity of an SVM implies that it will maximize the distance between the lines that separate classes, and therefore has a more sophisticated learning rule than LDA. We can arguably interpret a linear SVM as a more powerful and general form of LDA, with a more flexible and sophisticated learning rule. With this in mind, the first test we perform is based on linear SVM classification. The initial i-Vectors are still compressed by LDA to suppress non-accent information. It is the classifier that is changed from an LDA classifier to a linear SVM. The results for this test are shown in Figure 7.8.

The trend of obtaining better classification results with a larger number of components for

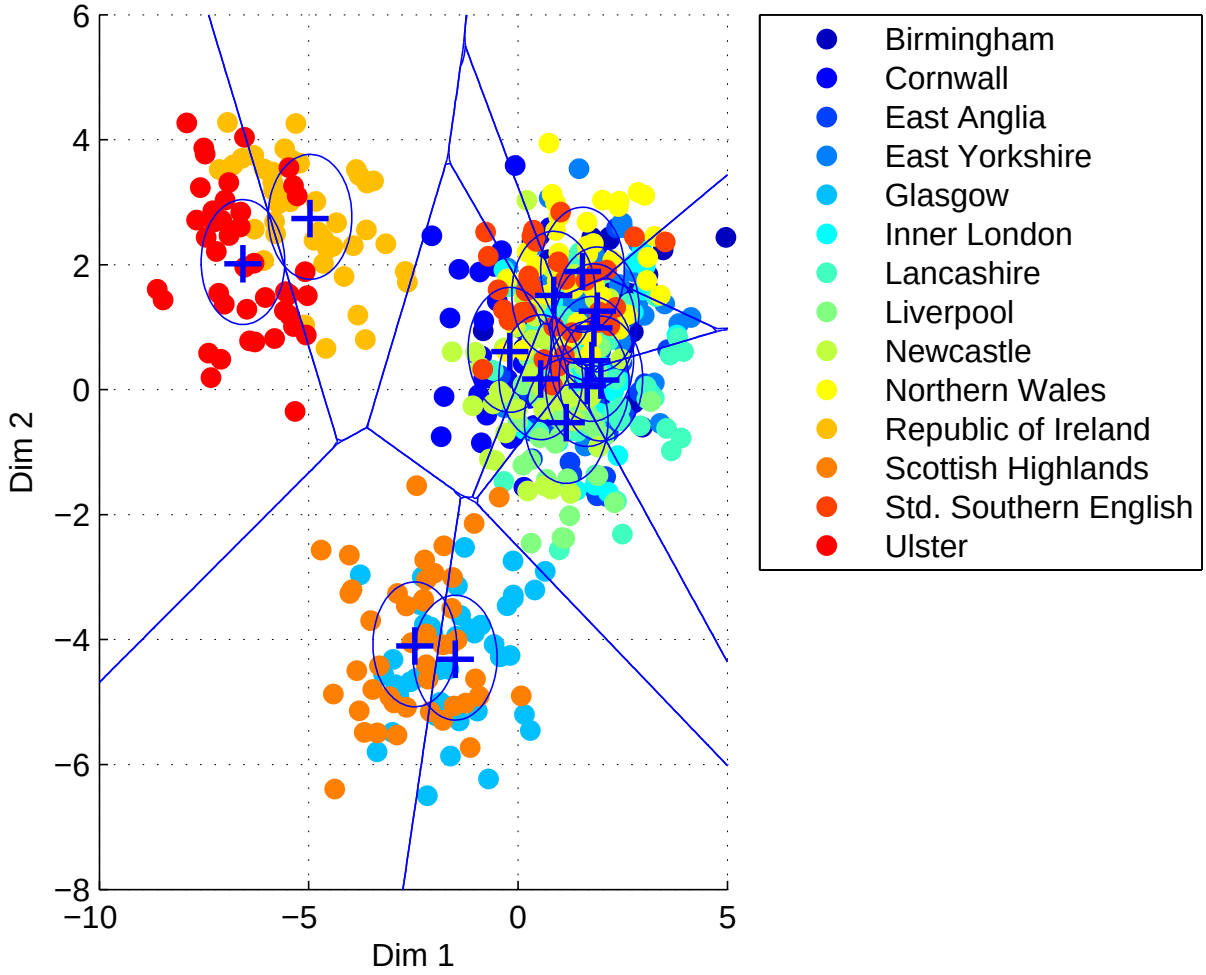


Figure 7.5: The i-Vectors are transformed by the LDA projection and are then classified with a LDA classification boundary. The first two dimensions of the data obtained by LDA reduction are shown.

the UBM continues in this case as well. Also, lower factor dimensions give better AID accuracy performance than larger factor dimensions. General performance however, is roughly the same for linear SVM classification and LDA classification. The apparently more powerful linear SVM classification does not solve the problem related of overlap of accent clusters. The AID accuracy obtained in this approach ranges from 42% to 73%. This equivalent performance is also surprising and suggests that the LDA dimensionality reduction in 13 dimensions (for 14 accents) may be putting an upper-limit restriction on classification performance.

The next trial makes use of a non-linear RBF kernel SVM classifier on the same data. The RBF kernel parameters are the default parameters from the LIBSVM toolkit. The results for this test are shown in Figure 7.9. The trend of obtaining better classification results with a larger number of components for the UBM continues in this case as well. In this case, the lower factor dimensions give better AID accuracy results as well. However, the higher factor dimensions

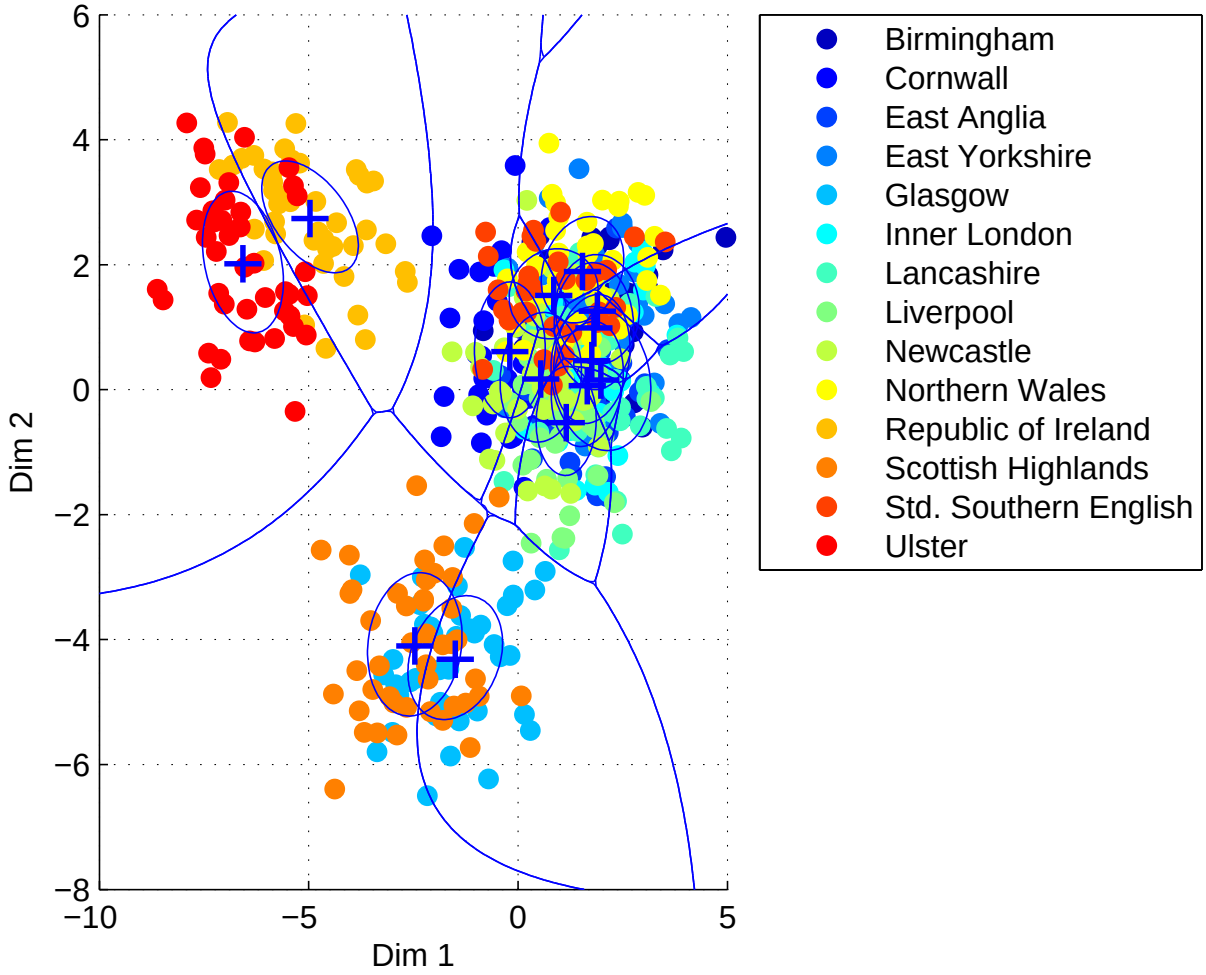


Figure 7.6: The i-Vectors are transformed by the LDA projection and are then classified with a QDA classification boundary. The first two dimensions of the data obtained by LDA reduction are shown.

give considerably very poor results when compared to other tests on i-Vector classification so far. For the lowest factor dimensionality of 100, performance is better than classification with a linear SVM . However, all other tests showed considerably poorer performance. Although the best SVM classifier obtains equivalent performance to the LDA classifier at 74%, the LDA system in general is very much ahead, as the AID accuracy for SVM classifiers ranges from 7% (equivalent to chance level) to 74%.

7.6 Iterative LDA/QDA Projection Optimization (Approach IX)

From the projections obtained by LDA on i-Vectors in Figure 7.5 we note that there is a noticeable separation between accents into three superclusters (a collection of smaller clusters). One cluster is formed by the Ulster and Republic of Ireland accents, another by the Scottish Highlands

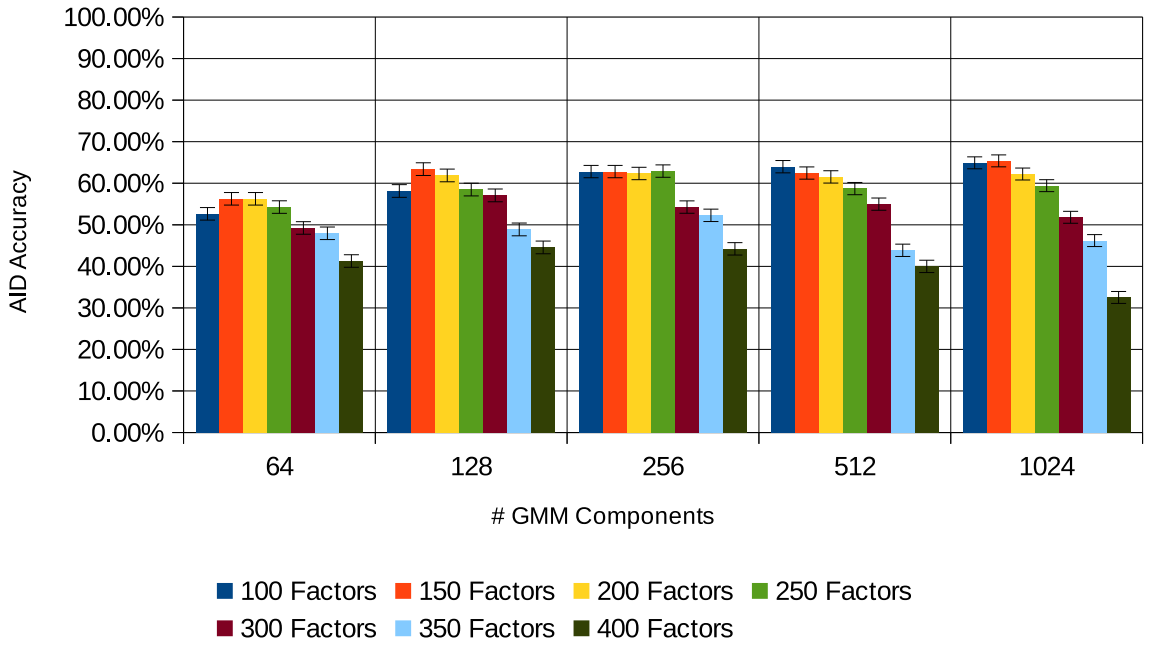


Figure 7.7: The second trial of the i-Vector paradigm on accent identification using QDA rather than LDA classification.

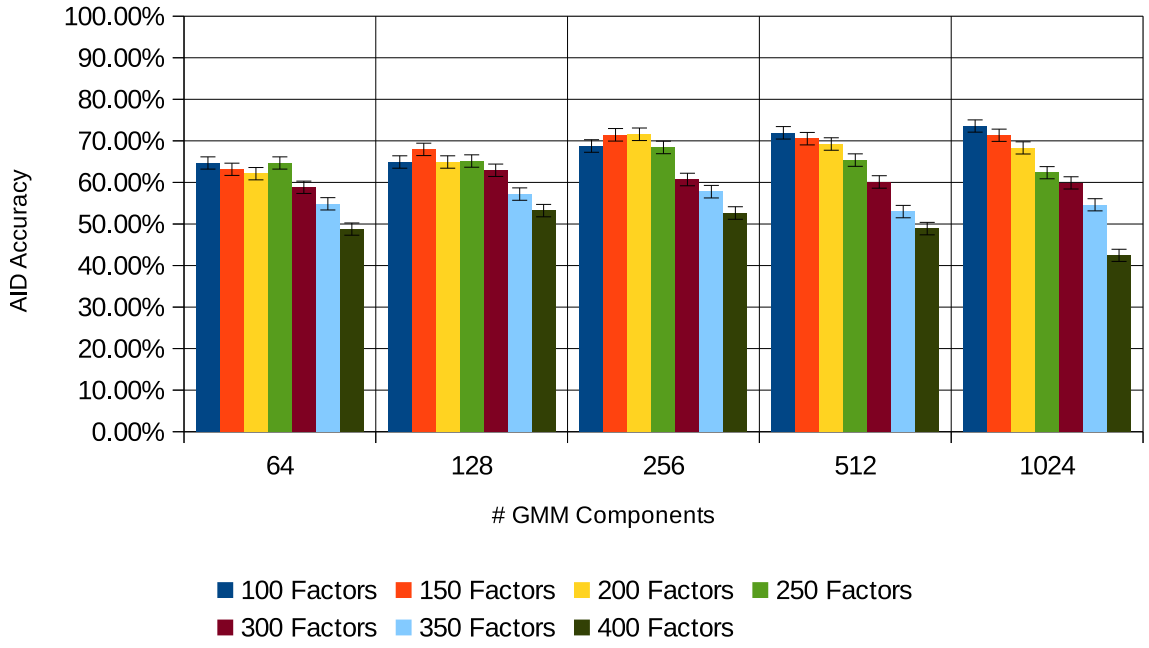


Figure 7.8: The third trial of the i-Vector paradigm on accent identification, using linear SVM classification on LDA-reduced i-Vectors.

and Glasgow accents and a final and larger cluster is formed by the other ten accents. There is a potentially problematic situation in this large supercluster of overlapping accents, which if resolved, could potentially boost the classification results we are obtaining. In Figure 7.10 we show the clusters of accent data from these ten highly overlapping clusters alone, after the LDA

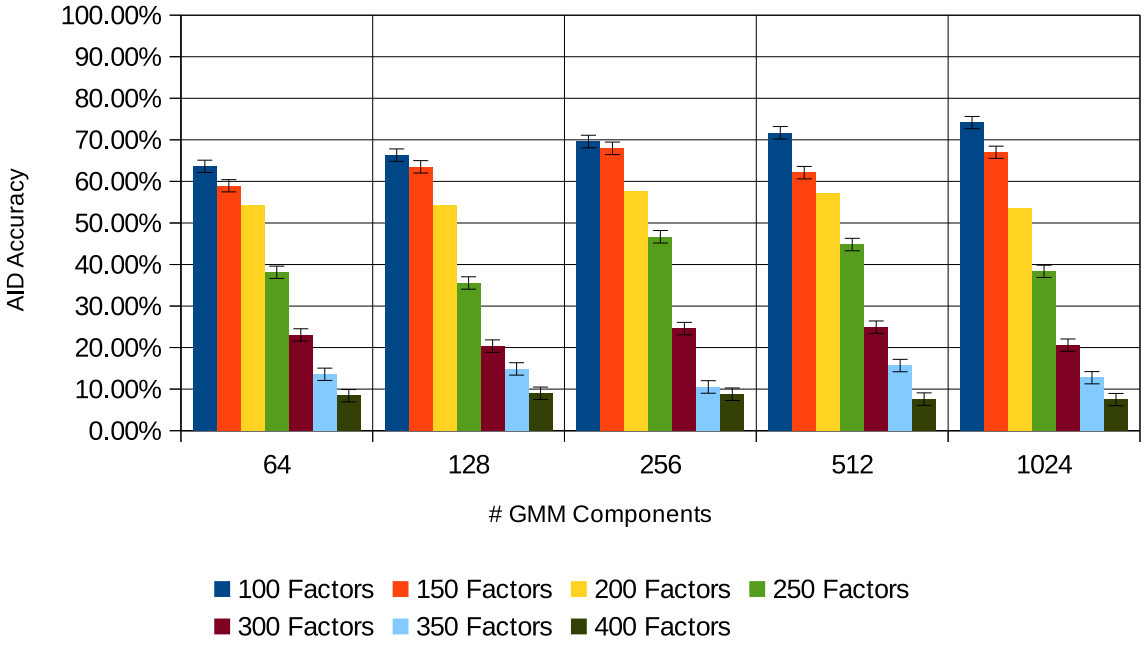


Figure 7.9: The third trial of the i-Vector paradigm on accent identification - RBF SVM classification on LDA-reduced i-Vectors.

projection is applied.

This detail of ten accents shows that there is some separation between classes, but the amount of overlap is too large for good accent classification. If we assume for a moment, that we only had to perform accent classification on these ten classes, then the LDA projection we obtain would have been different. This is shown in Figure 7.11. It is evident that the spread between classes has improved, as by reducing the original amount of data and classes to the LDA algorithm, we have provided it with a way to maximize class separation even further.

With this in mind, we can argue that during test time, we can use the rough location of the LDA-projected i-Vector of an utterance to roughly assess the zone of interest, rather than to do direct classification. This zone of interest might allow us to re-optimize the LDA projection for only a subset of classes which are within this zone of interest. We propose a novel classification framework (first demonstrated in [9]) based around two generative classification methods: one LDA, and another based on QDA. Classification of a test utterance proceeds as follows:

1. Obtain initial LDA classifier L^* and QDA classifier Q^* using all accent classes.
2. $L \leftarrow L^*$, $Q \leftarrow Q^*$
3. Classify test utterance using L and Q and rank classes in order of likelihood. Identify lowest ranking class and remove it from training data.

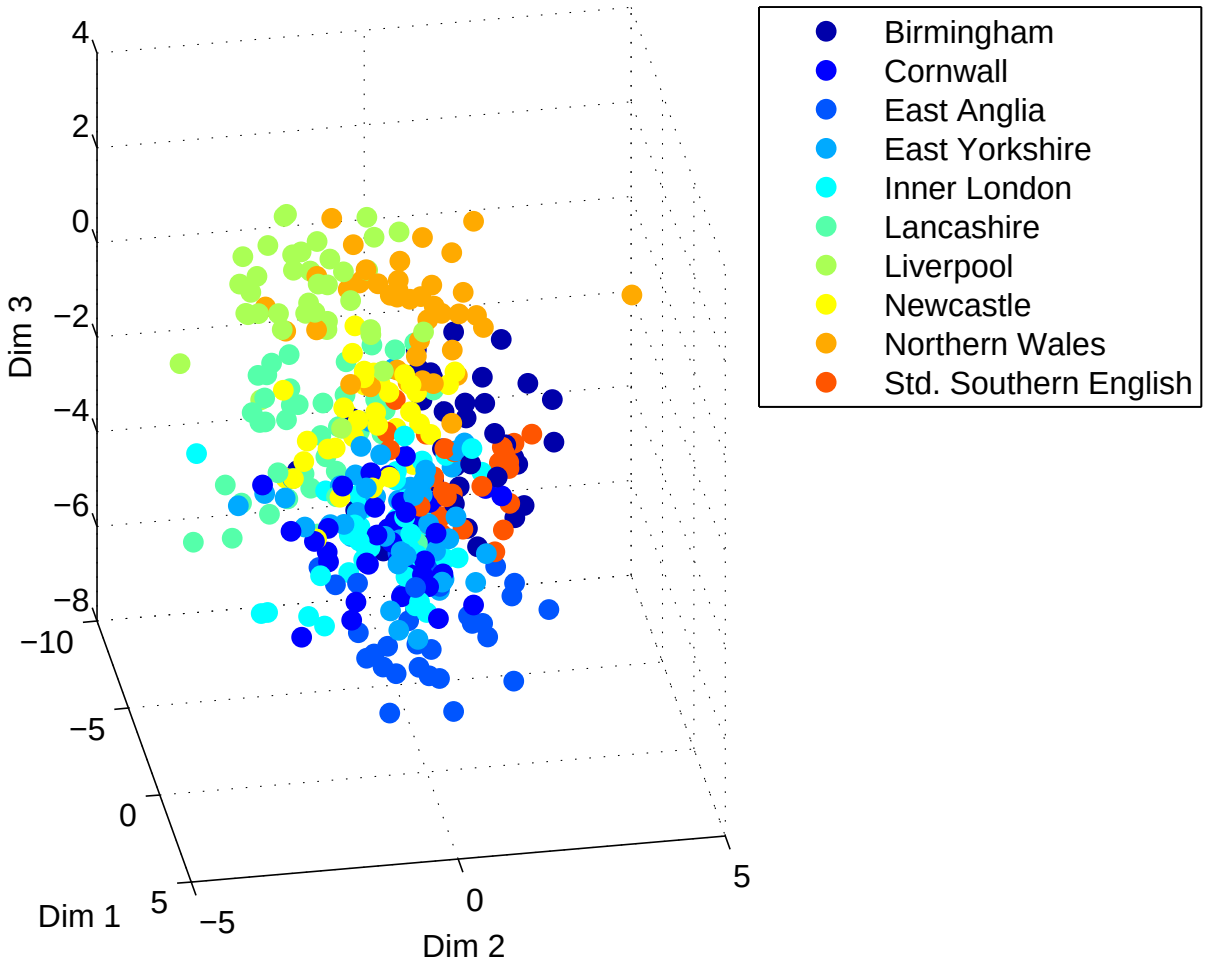


Figure 7.10: A large supercluster (or collection of clusters) of 10 accents out of the original 14 from the original LDA projection over all i-Vectors from 14 accents. The first three dimensions of the data obtained by LDA reduction are shown.

if a single class remains in the training data **then**

Classify utterance (see below)

else

Re-train LDA/QDA classifiers using reduced training data

$L \leftarrow L^*$, $Q \leftarrow Q^*$

Goto 3 with new classifiers L and Q

end if

Traditional LDA/QDA classification would produce a result after the first scoring, by selecting the class with the highest likelihood. However, by applying the above iterative algorithm, we remove at an early stage classes that are likely to be incorrect (not in the zone of the test utterance), and hence strengthen the accumulation of evidence for classes that appear to be good contenders for the correct class. Because each iteration of the algorithm removes a class, the vector dimensionality reduces by one on each iteration, which is another bonus in this

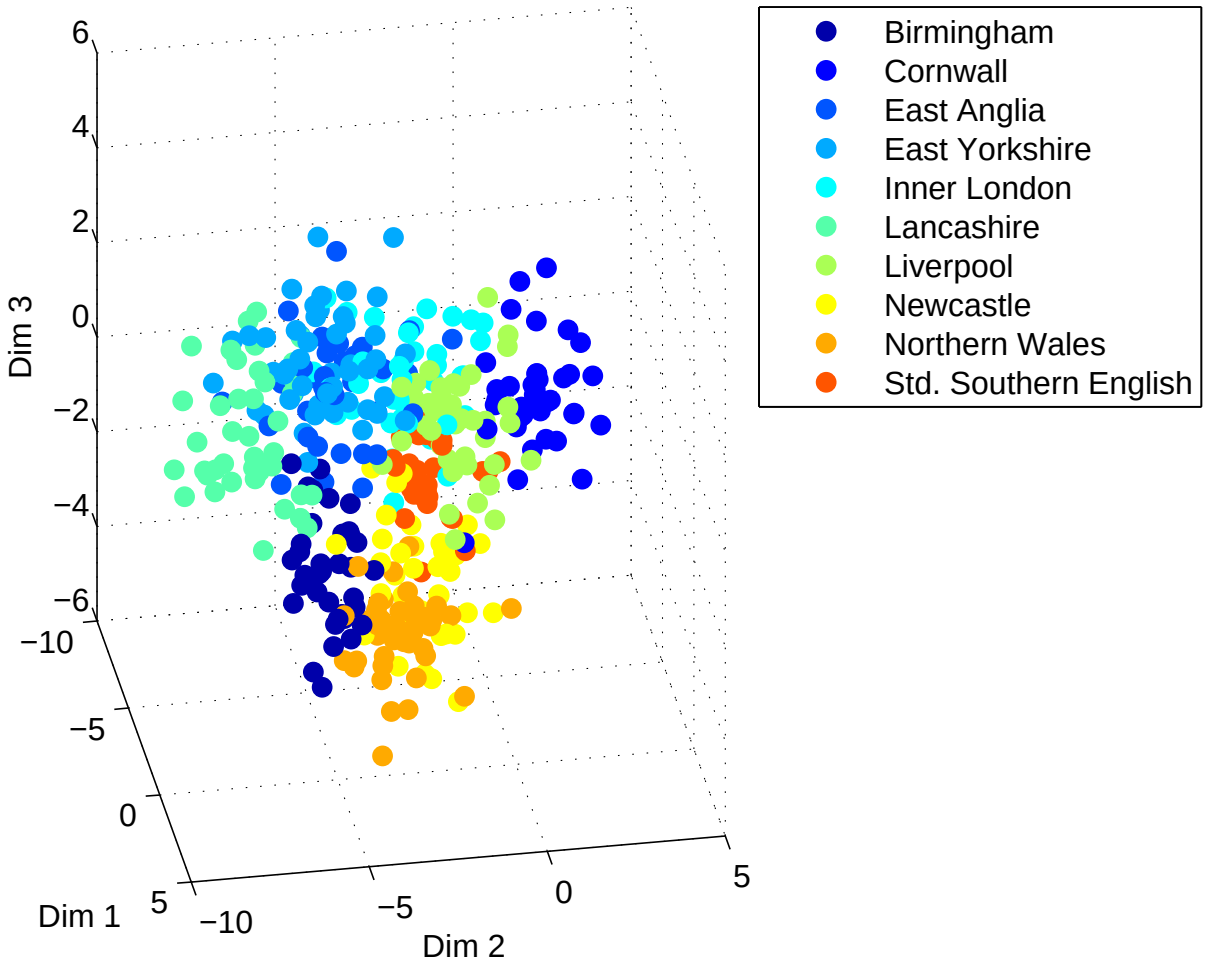


Figure 7.11: A large supercluster (or collection of clusters) of 10 accents with a specific LDA projection obtained from i-Vectors from only 10 out of the 14 original accent classes. The first three dimensions of the data obtained by LDA reduction are shown.

technique.

The motivation for this iterative approach is that it could help to eliminate the larger acoustic overlap between classes in accent identification for the highly overlapping accent classes. The proposed algorithm attempts to iteratively sharpen the separation between classes by removing the weakest candidates at each iteration: these classes contribute mainly noise to the classification process. The rank of each class and the order in which classes are removed is recorded for the test utterance. Two possible ways in which this information could be used are:

- Classification method 1: the last class to be eliminated is the classification result, or
- Classification method 2: the class that had the best (top rank) likelihood for most iterations is the classification result

We can consider a few classification examples to put these methods into perspective. Three

Table 7.1: This table shows three examples (starting at columns two, five and eight respectively) of the iterative classification procedure working. For each example, the target class is given in the first column, the second columns shows the ranked position of the target class and the third column shows the identity of the class removed at each iteration.

Iteration	Target (Real Accent)	Rank of Target	Class Removed	Target (Real Accent)	Rank of Target	Class Removed	Target (Real Accent)	Rank of Target	Class Removed
1	ULS	4	shl	ULS	1	shl	CRN	6	ean
2	ULS	3	ean	ULS	1	ilo	CRN	6	shl
3	ULS	3	lan	ULS	1	nwa	CRN	6	uls
4	ULS	2	crn	ULS	1	ean	CRN	5	gla
5	ULS	4	ilo	ULS	1	eyk	CRN	5	roi
6	ULS	3	eyk	ULS	1	lan	CRN	5	lan
7	ULS	3	lvp	ULS	1	crn	CRN	4	eyk
8	ULS	1	ncl	ULS	1	lvp	CRN	5	brm
9	ULS	3	nwa	ULS	1	ncl	CRN	4	ncl
10	ULS	1	sse	ULS	1	gla	CRN	4	lvp
11	ULS	1	gla	ULS	2	brm	CRN	3	sse
12	ULS	1	brm	ULS	2	sse	CRN	3	crn
13	ULS	1	roi	ULS	2	uls	CRN	-	nwa
14	ULS	1	uls	ULS	-	roi	CRN	-	ilo

examples of the classification processes are shown in Table 7.1. The first example shows a case where the target rank gradually climbs to one as incorrect classes are removed by the LDA/QDA classifier during the iterative procedure. The second example is a case where the final classification would be incorrect under the first classification technique, but is correct using the second. The third example is a case where classification is incorrect under both classification methods. However, we can see that the iterative procedure has still managed to increase the target class rank during the iterations.

Results are shown in Figures 7.12 and 7.13. Both classification decision methods perform well, but the second classification method gives better performance in all conditions. The range of AID accuracy for classification method 1 is at 36% to 76%, whilst the range of AID accuracy for classification method 2 is at 40% to 78%. Both of these techniques produce better AID accuracy than standard LDA when selecting the best performing configuration. However, the worst classifier for classification method 1 performs worse than standard LDA classification.

With the utilization of QDA classification giving slightly worse performance than with LDA classification, we expect the same from the iterative counterpart as well. The second set of results in Figures 7.14 and 7.15, shows iterative QDA classification for both classification decision methods.

For both classification decision methods, performance of iterative QDA is better than

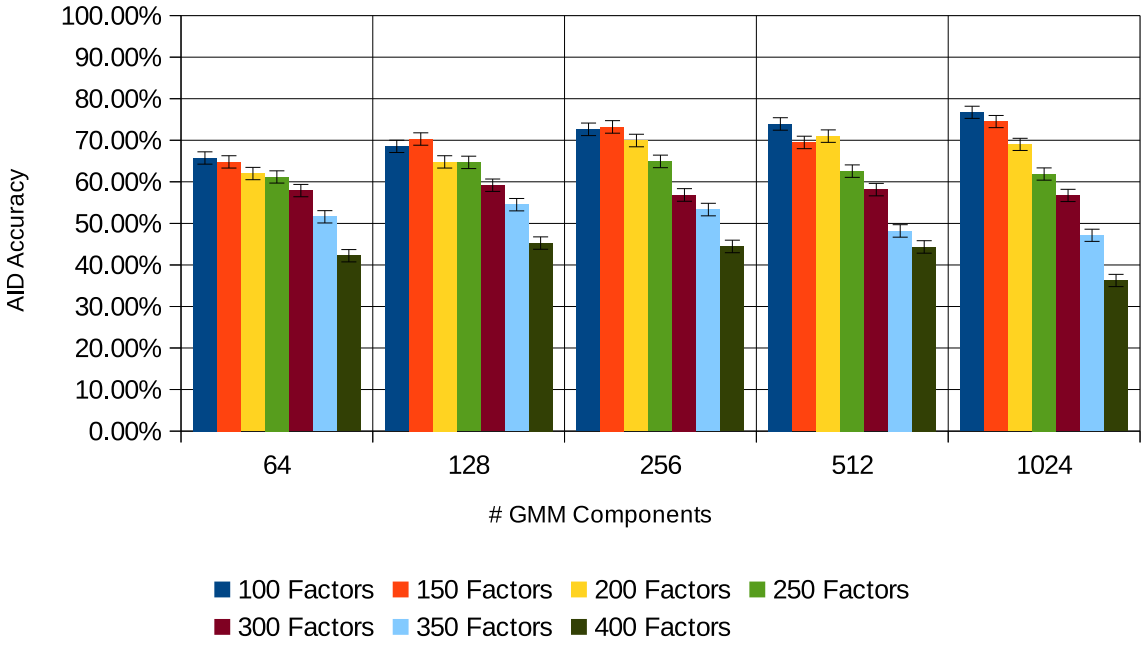


Figure 7.12: Performance of iterative LDA using classification method 1.

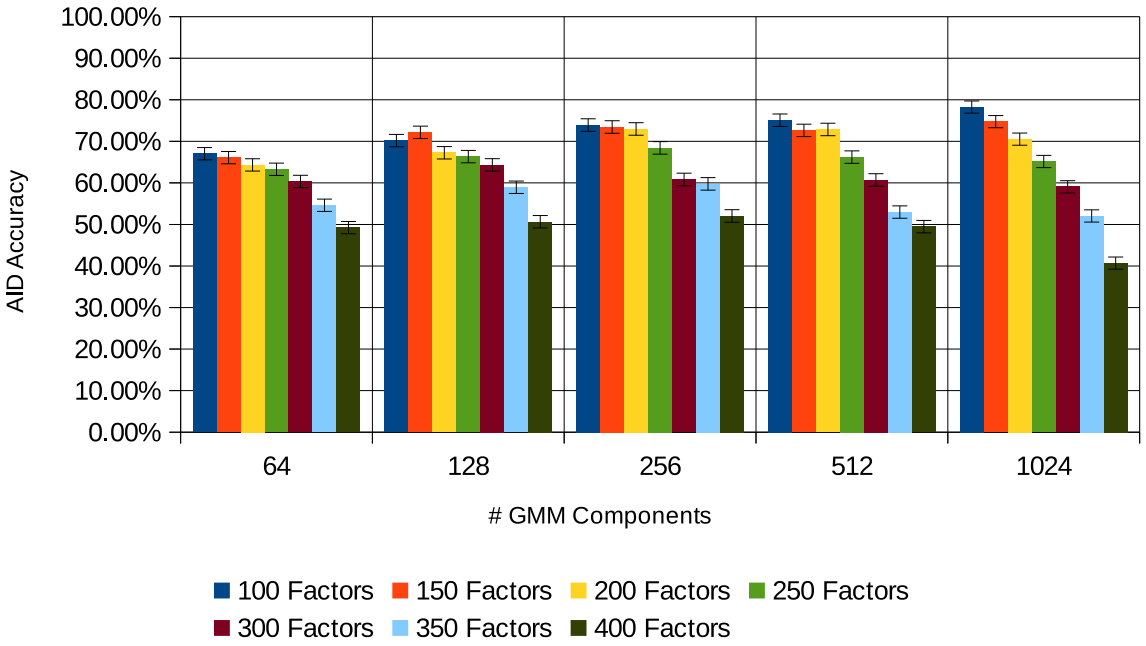


Figure 7.13: Performance of iterative LDA using classification method 2.

standard QDA classification. Iterative LDA achieves overall better AID accuracy. For both iterative LDA and iterative QDA, the trend of better accuracy with more GMM components and low factor dimensions continues as with standard LDA and QDA, as well as SVM classification. The AID accuracy range for the first classification method is at 28% to 67%, whilst for the second classification method, performance ranges from 30% to 66%. As with the non-iterative

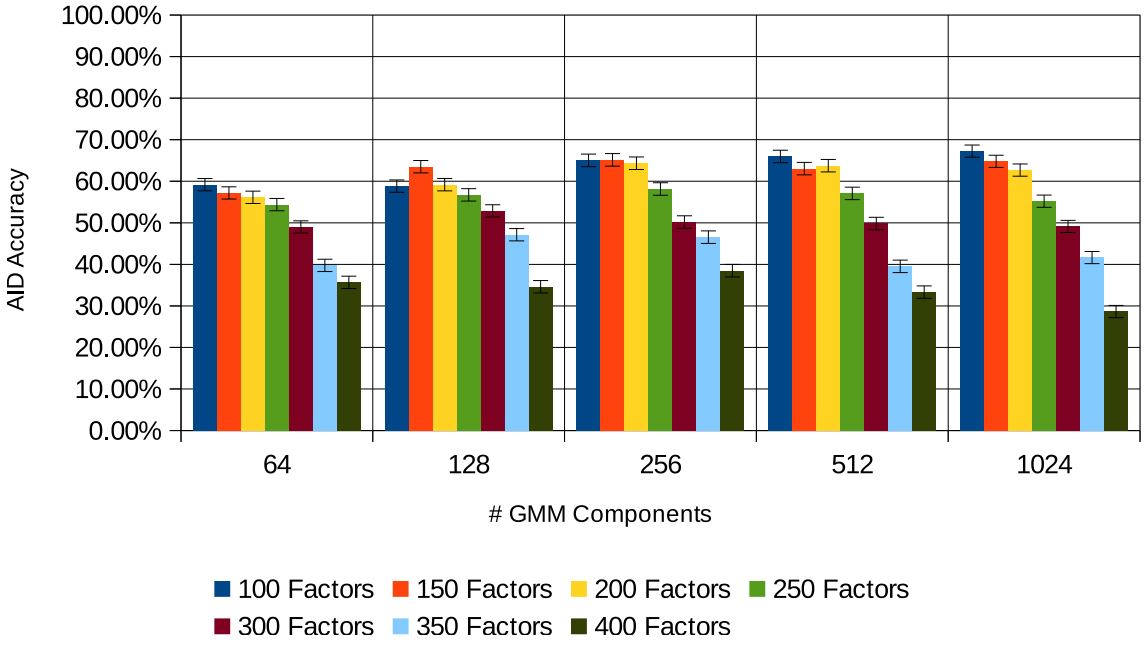


Figure 7.14: Performance of iterative QDA using classification method 1.

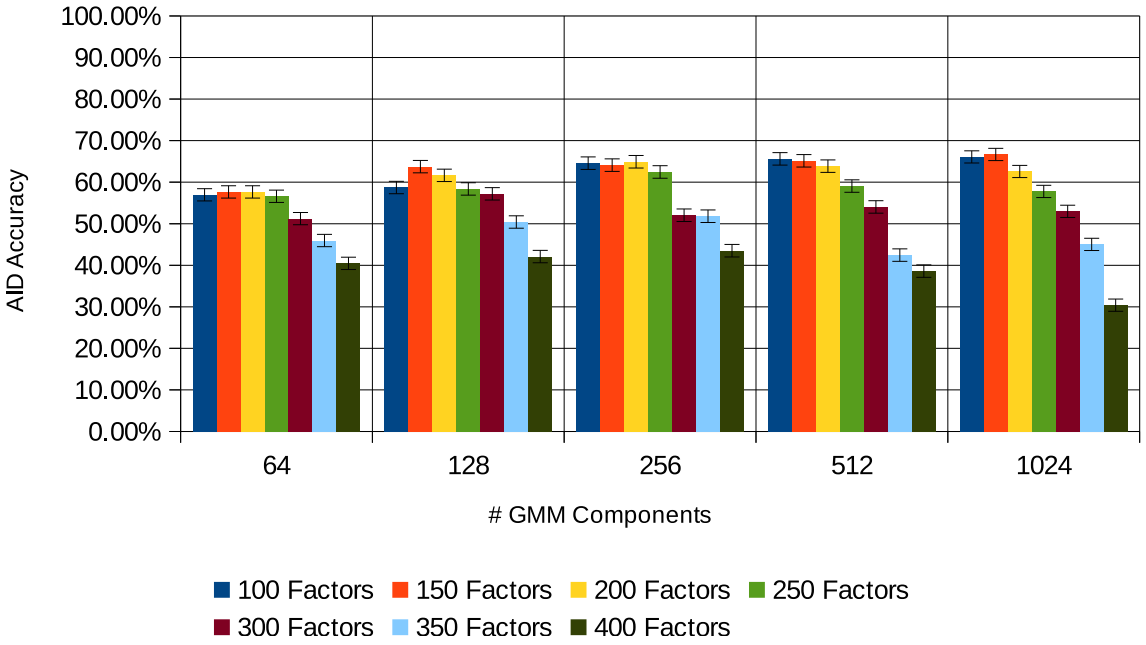


Figure 7.15: Performance of iterative QDA using classification method 2.

counterparts, LDA is better suited to this classification problem.

We find the results of these experiments encouraging. Given the same i-Vectors for all experiments prior to dimensionality reduction and classification, we can observe that the projection and eventual classification have a bearing on the final classification result. We can also construct a simple confidence measure using output from the iterative classification algorithm

which should be useful in any techniques that integrate decisions from different classifiers. For a given test utterance, let N_C be the number of times the *correct* class was top-ranked at each iteration, which is a maximum of 14 (the number of classes), and a minimum of zero. Figure 7.16 shows the distribution of a confidence measure, $CM = N_C/14$, for correctly classified utterances and incorrectly classified utterances. Although there is some overlap between the two distributions, it is clear that higher values of CM are correlated with correct classification, and this encourages us that CM will be useful in fusion with other classifiers.

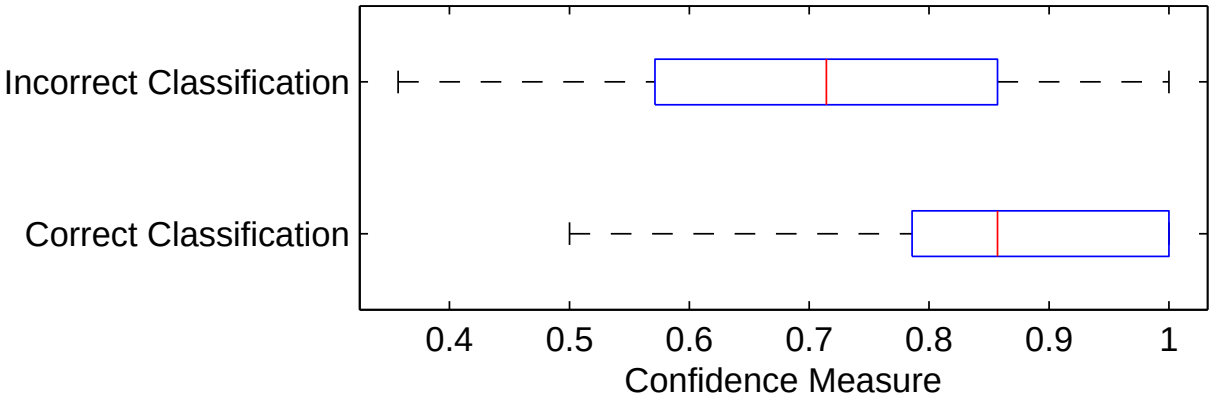


Figure 7.16: Box and whisker plot for confidence measure.

A more detailed breakdown of the distribution of confidence measures is shown in Figure 7.17. The distributions are very different. For incorrect classification, we can observe a quasi-normal distribution around a mean of 0.7 confidence. On the other hand, for incorrect classification, the distribution is heavily weighted towards higher confidence values growing exponentially from a minimum of 0.5 to a maximum of 1.0 confidence. It is clear that using this confidence measure, for most of the correct classifications, the algorithm provides very high confidence, very regularly. The opposite is true for those utterances resulting in incorrect classification.

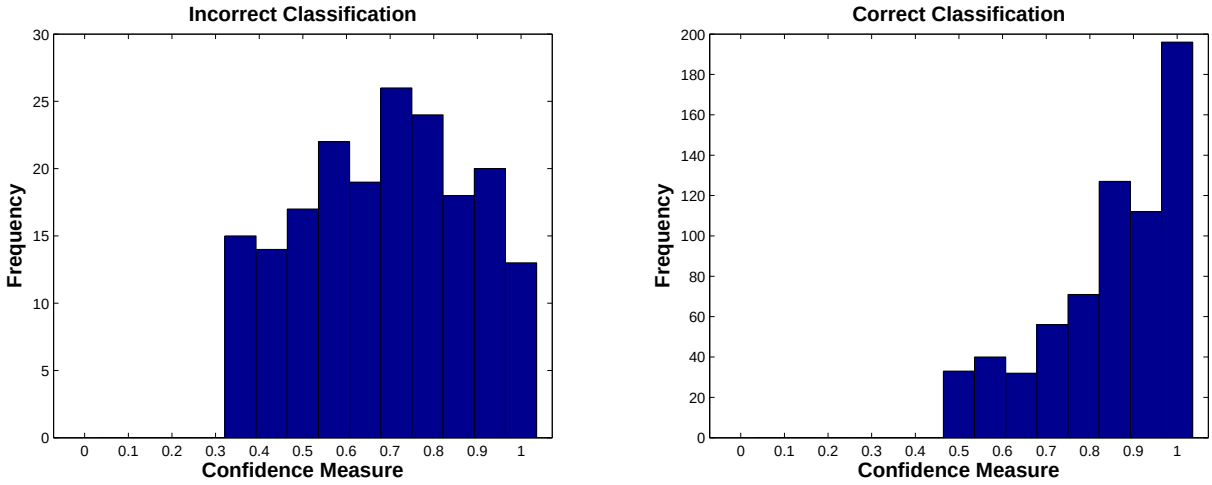


Figure 7.17: Histograms of confidence measures for iterative discriminant analysis classification.

7.7 Accent Confusion Analysis (Part 1)

We shall now have another look at which accents are being confused with others and evaluate whether the i-Vector paradigm gives better insight into whether some accents are harder or easier to classify, and in the case of wrongly classified accents, whether the chosen wrong classification has some acoustic/phonetic basis. A confusion matrix showing the correct versus predicted classification of utterances of the corpus for Approach IX is shown in Table 7.4. A list of closest confusions per accent is given in Table 7.2. This presents a very different picture to the results we observed in Section 6.6. The i-Vector paradigm provides some associations that are worth pointing out in the way accents are classified when comparing the classifications to the broad accent geographic formation (see Section 4.2).

Table 7.2: Ordered list of closest confusions per each accent as given by the Approach IX classifier. Where no confusions are made, columns are left empty.

Accent	Closest Accents												
brm	ilo	sse	ean	lan	lvp								
crn	ilo	sse	eyk	brm	ean	ncl							
ean	crn	eyk	sse	brm	ilo	ncl							
eyk	lan	ilo	sse	nwa	brm	crn	gla	ncl					
gla	shl	eyk	ilo	lan	ncl								
ilo	crn	eyk	brm	ean	sse	nwa	shl						
lan	eyk	brm	nwa	gla	lvp	ncl	sse						
lvp	nwa	brm	crn	ilo	ncl								
ncl	nwa	eyk	sse	ilo	lan	lvp							
nwa	ncl	brm	sse	crn	eyk	lan							
roi	uls	brm											
shl	gla												
sse	ean	ilo	brm	crn	eyk	roi							
uls	roi	brm	gla	sse									

The Birmingham accent (North, Midlands) has been confused with the North Wales accent, which is a northern (albeit independently Welsh) accent. The Cornwall accent (South, South West) has been confused with the Inner London accent (South, London). The Glasgow accent is confused with the Newcastle accent - and both are geographically far North. Lancashire and East Yorkshire are both North to Mid-North accents and have been confused together. The same can be said for the Liverpool accent confused with both the North Wales and the Birmingham accents. The Newcastle accent utterances are confused with the North Wales accent, and the Scottish Highlands accent is confused only with the Glasgow accent (the closest geographic region to it). Standard Southern English is confused mainly with other Southern accents like the East Anglia and Inner London accents. The Republic of Ireland accents is confused with Ulster

(Northern Irish) accent, and followed by North Wales. Ulster in turn is also confused back with the Republic of Ireland accent. There are other accents that are confused in a way that does not make much sense in terms of how the accents are mapped across the British Isles. However for most accents, the primary confusion seems to be from accents that can be considered "similar" and "related".

Table 7.3: Ordered list of closest accents for every other accent as given by the Approach IX training results.

Accent	Closest Accents												
brm	ilo	ean	sse	ncl	nwa	eyk	lan	crn	lvp	gla	roi	uls	shl
crn	ilo	sse	ncl	nwa	eyk	ean	brm	lvp	roi	lan	shl	gla	uls
ean	brm	ilo	sse	eyk	ncl	crn	lan	nwa	lvp	roi	gla	uls	shl
eyk	lan	ilo	ean	nwa	sse	ncl	brm	crn	roi	lvp	gla	uls	shl
gla	shl	ilo	ncl	eyk	lan	lvp	uls	crn	roi	sse	nwa	brm	ean
ilo	sse	crn	brm	eyk	ean	ncl	lan	nwa	lvp	gla	roi	uls	shl
lan	eyk	ilo	ncl	brm	nwa	sse	ean	lvp	crn	gla	roi	uls	shl
lvp	nwa	ncl	ilo	sse	eyk	lan	crn	brm	gla	ean	roi	shl	uls
ncl	nwa	brm	ilo	crn	sse	eyk	lan	lvp	ean	gla	roi	shl	uls
nwa	ncl	sse	eyk	brm	lvp	crn	lan	ilo	ean	roi	gla	uls	shl
roi	uls	sse	eyk	crn	ncl	nwa	ilo	gla	brm	ean	lvp	lan	shl
shl	gla	crn	ncl	eyk	ilo	lvp	sse	nwa	roi	lan	ean	uls	brm
sse	ilo	brm	nwa	crn	ean	eyk	ncl	roi	lan	lvp	gla	uls	shl
uls	roi	gla	sse	ncl	eyk	ilo	crn	nwa	brm	ean	lan	lvp	shl

Another analysis which can be looked at is the proximity of accents to each other after the training phase is completed, through the Euclidean distance of the mean of each accent cluster. This is shown in Table 7.3. Even here, the accent mapping learnt in the training phase closely matches the confusions in the classification stage for 12 of the 14 accents. The disagreements occur for the Standard Southern English accent, which is closest to the Inner London accent as opposed to most confusions having been with the East Anglian accent, and the East Anglia accent which is (strangely) closest to the Birmingham accent as opposed to most confusions having been with the Cornwall accent. Despite some inconsistencies, this is overall a big step ahead both in classification results, and on the intuitive accent mapping that the i-Vector paradigm has learnt from the training data. The iterative classification techniques we proposed in this section have given some measure of improvement over standard algorithms. All other conditions being equal, improving the LDA stage of the i-Vector framework seems to be a crucial step, and in the next section we propose a different idea to improve results. The work in the last section also described the first application of the i-Vector paradigm, together with our own improvements, on the problem of accent identification. As with all other similar speech classification problems like speaker and language identification, the i-Vector paradigm

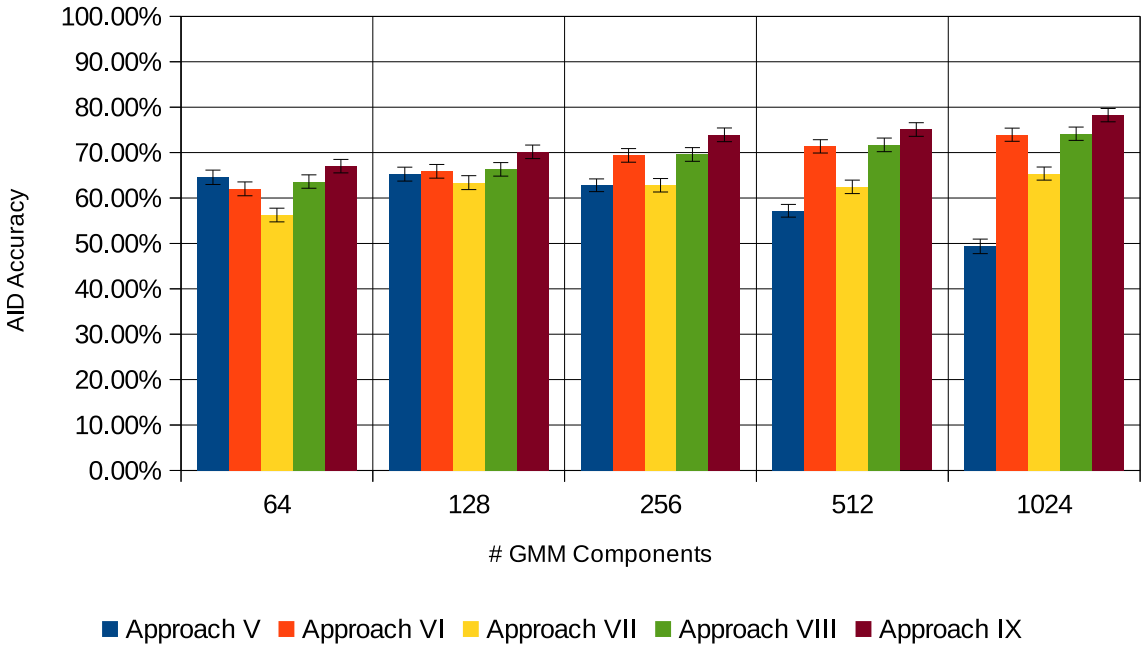


Figure 7.18: Comparison of AID performance for the best configurations under different classification techniques: Approach V (RBF kernel SVM after PCA+LDA dimensionality reduction), Approach VI (i-Vector classification via LDA projection and LDA classifier), Approach VII (i-Vector classification via LDA projection and QDA classifier), Approach VIII (i-Vector classification via LDA projection and linear SVM classifier), Approach IX (i-Vector classification via LDA projections and iterative LDA classification).

also improves results for accent identification over previous methods. A final comparison of performance results across the best approaches prior to the utilization of the i-Vector paradigm, with the i-Vector based methods is shown in Figure 7.18.

Table 7.4: Confusion matrix (in %) of correct vs predicted classification of utterance for the 14 accents of the British Isles for Approach IX. Average accent recognition accuracy is of 78.25%. The diagonal, which represents the correct (no confusion) results is in **bold**, whilst off-diagonals (confusion) equal or greater than 8% are marked in **red**.

	brm	crn	ean	eyk	gla	ilo	lan	lvp	ncl	nwa	roi	shl	sse	uls
brm	**73.33**	0.00	5.00	0.00	0.00	**8.33**	3.33	3.33	0.00	0.00	0.00	0.00	6.67	0.00
crn	1.67	**73.33**	1.67	5.00	0.00	**8.33**	0.00	0.00	1.67	0.00	0.00	0.00	**8.33**	0.00
ean	1.75	5.26	**78.95**	5.26	0.00	1.75	0.00	0.00	1.75	0.00	0.00	0.00	5.26	0.00
eyk	2.67	2.67	0.00	**66.67**	1.33	5.33	**10.67**	0.00	1.33	4.00	0.00	0.00	5.33	0.00
gla	0.00	0.00	0.00	1.67	**88.33**	1.67	1.67	0.00	1.67	0.00	0.00	5.00	0.00	0.00
ilo	3.17	6.35	3.17	4.76	0.00	**76.19**	0.00	0.00	0.00	1.59	0.00	1.59	3.17	0.00
lan	7.94	0.00	0.00	**11.11**	1.59	0.00	**71.43**	1.59	1.59	3.17	0.00	0.00	1.59	0.00
lvp	1.67	1.67	0.00	0.00	0.00	1.67	0.00	**85.00**	1.67	**8.33**	0.00	0.00	0.00	0.00
ncl	0.00	0.00	0.00	3.33	0.00	1.67	1.67	1.67	**80.00**	**8.33**	0.00	0.00	3.33	0.00
nwa	4.76	3.17	0.00	1.59	0.00	0.00	1.59	0.00	**14.29**	**69.84**	0.00	0.00	4.76	0.00
roi	3.33	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	**91.67**	0.00	0.00	5.00
shl	0.00	0.00	0.00	0.00	3.03	0.00	0.00	0.00	0.00	0.00	0.00	**96.97**	0.00	0.00
sse	6.25	4.17	**8.33**	4.17	0.00	**8.33**	0.00	0.00	0.00	0.00	2.08	0.00	**66.67**	0.00
uls	1.67	0.00	0.00	0.00	1.67	0.00	0.00	0.00	0.00	0.00	**8.33**	0.00	1.67	**86.67**

7.8 The Effect of i-Vector Length Normalization

Length normalization over i-Vectors is reported to give performance gains in some classification tasks [210]. This process is performed by normalizing every i-Vector to a unit vector. The reason for doing so is that the i-Vectors may be exhibiting non-Gaussian behaviour, and the non-linear transformation in length normalization allows for better used of probabilistic modelling that has Gaussian assumptions such as the UBM followed by i-Vector extraction paradigm. In principle, one expects an i-Vector extractor to produce vectors that have a normal distribution. However the work in [210] has observed that it is common to have length mismatches from an i-Vector extractor given development and test data that is different in some respects.

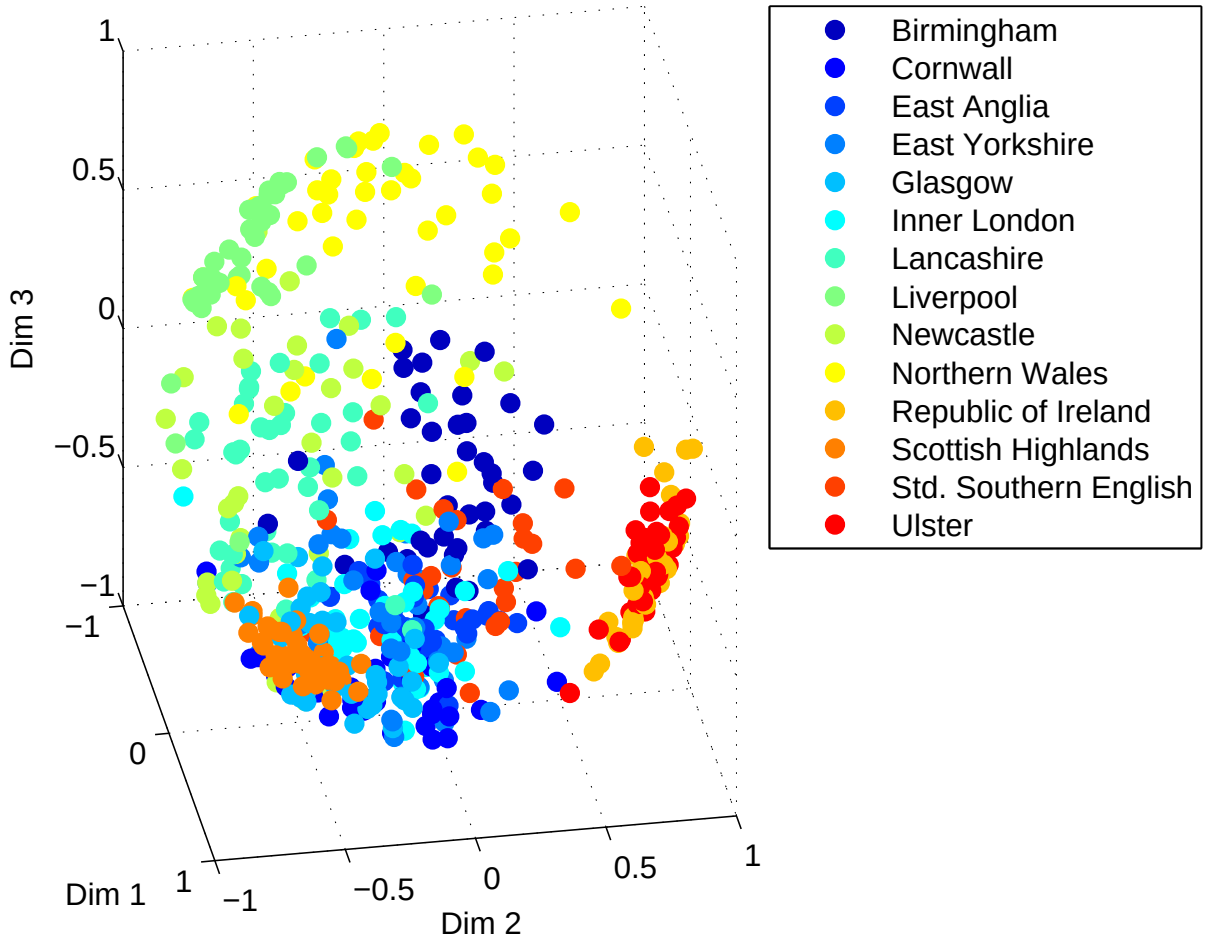


Figure 7.19: Length normalization of i-Vectors after LDA dimensionality reduction was applied. The first three dimensions of the data obtained by LDA reduction are shown.

We perform an analysis of the effect of i-Vector Length Normalization in our experiments as well. In the first trial, we extract the i-Vectors as before, perform length normalization on the i-Vectors, then perform LDA dimensionality reduction, and then proceed to LDA classification. In the second trial, we extract the i-Vectors, perform LDA dimensionality reduction first,

followed by length normalization, and then proceed to LDA classification. An example of length-normalized i-Vectors on the unit sphere is shown in Figure 7.19. The results of these trials are shown in Figures 7.20 and 7.21 respectively.

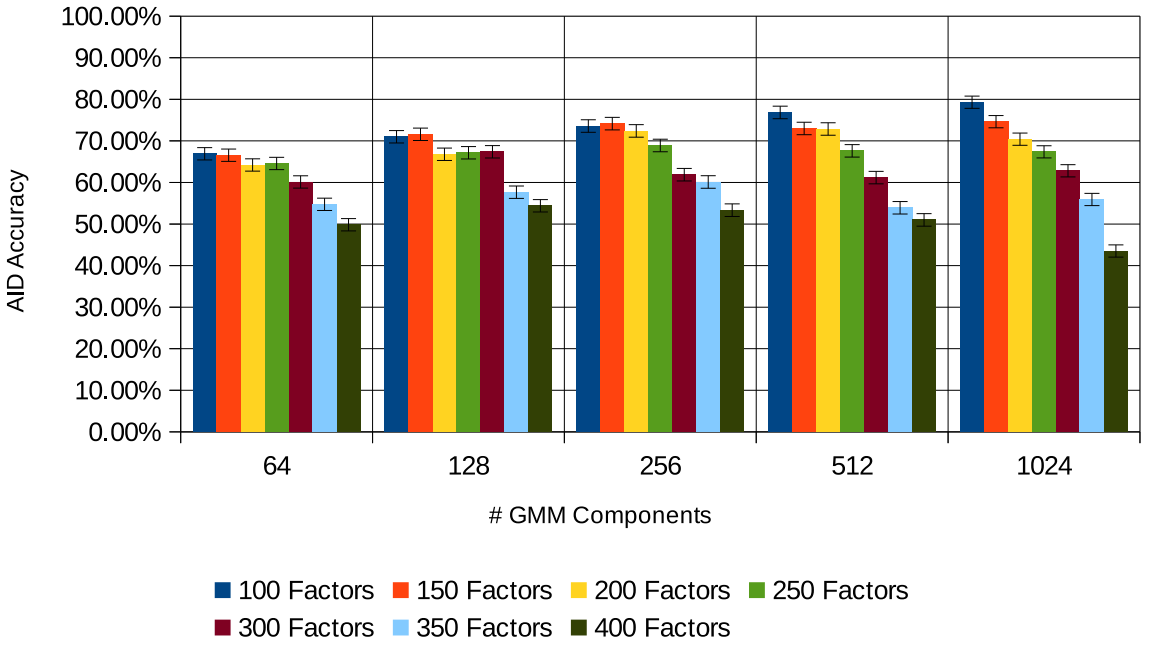


Figure 7.20: AID classification accuracy for i-Vectors that have been first length normalized and then projected to a lower dimensionality via LDA. Classification is performed via non-iterative LDA.

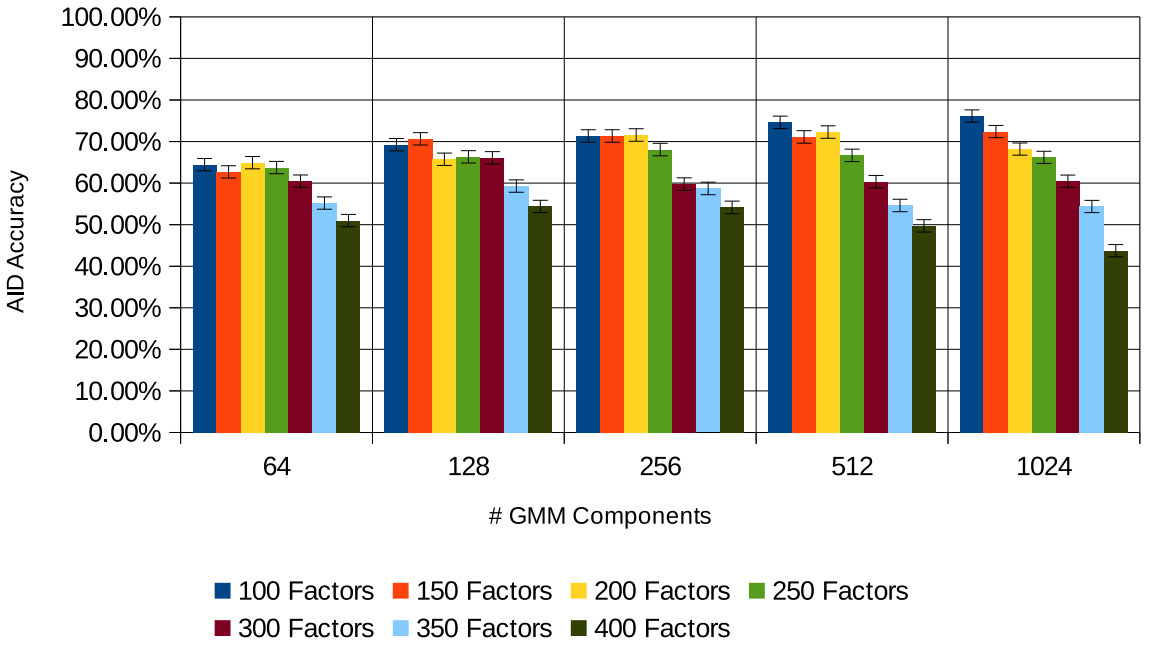


Figure 7.21: AID classification accuracy for i-Vectors that have been first projected to a lower dimensionality via LDA and then length normalized. Classification is performed via non-iterative LDA.

There are a number of interesting results arising from this experiment. The results in Figure 7.20 show that AID accuracy performance falls in the range of 43% to 79%, whilst the results in Figure 7.20 show results in the range of 43% to 76%. Both of these results are better than those obtained in the original LDA experiment in Approach VI, which had results in the range of 39% to 74%. The iterative-LDA procedure with the second mode of classification in Approach IX produced AID accuracy results in the range of 40% to 78%. This means that length normalization prior to LDA dimensionality reduction over standard LDA obtains even better results than the iterative LDA technique. The total improvement on the best configuration is of only 1%. When we compare the results with the original LDA technique, we realize that length normalization gives an improvement of 5%, which is quite significant.

7.9 Speaker Compensation Fusion (Approach X)

There seems to be something in the various projections given by a learnt transformation with a number of assumptions that limits the overall class-learning ability. A number of these assumptions may be incorrect. The arbitrary choices of the number of factors is also undesirable, even though we have identified better performance from lower factor numbers. In a new approach, we propose a way of fusing the different systems (built from a different number of GMM components and different factor dimensions) to increase performance.

If we consider one form of classification, say LDA classification, then we have 35 possible classifiers output from five GMM orders and the seven different factor dimensions for each order. We can consider a simple fusion mechanism where for the 35 classifier outputs, we take the majority class label as the final classification. But it may be that some classifiers are better than others in classifying utterances, and perhaps there is a special combination that gives the best possible classification output. To find the optimal solution, trying out $\binom{35}{1}+\binom{35}{2}+\ldots+\binom{35}{35}$ combinations of classifiers would be required, and this takes too long. For this reason we employ a binary genetic algorithm (GA) to find a quasi-optimal set of classifiers (although it is not known if this is the optimal combination). Each "chromosome" is a binary vector of 35 entries, with each binary entry indicating whether a particular classifier output should be considered in the majority vote (1) or not (0). The initial population is of 5000 individuals, with a generation gap of 0.9, a crossover rate of 0.5, and a mutation rate of 0.0175. The GA runs for 100 generations. The scoring function simply ranks each individual with the accuracy obtained by the fusion of the particular set of classifiers selected by that individual. All i-Vectors are

length normalized prior to LDA dimensionality reduction and classification. By the end of the GA optimization, the best classifier selection is compared with our previous results where no fusion was involved. These results are shown in Figure 7.22. The fused result is better than all other singular classifiers, and the performance of the best single classifier is improved by 7%, from 79% to 86% AID accuracy.

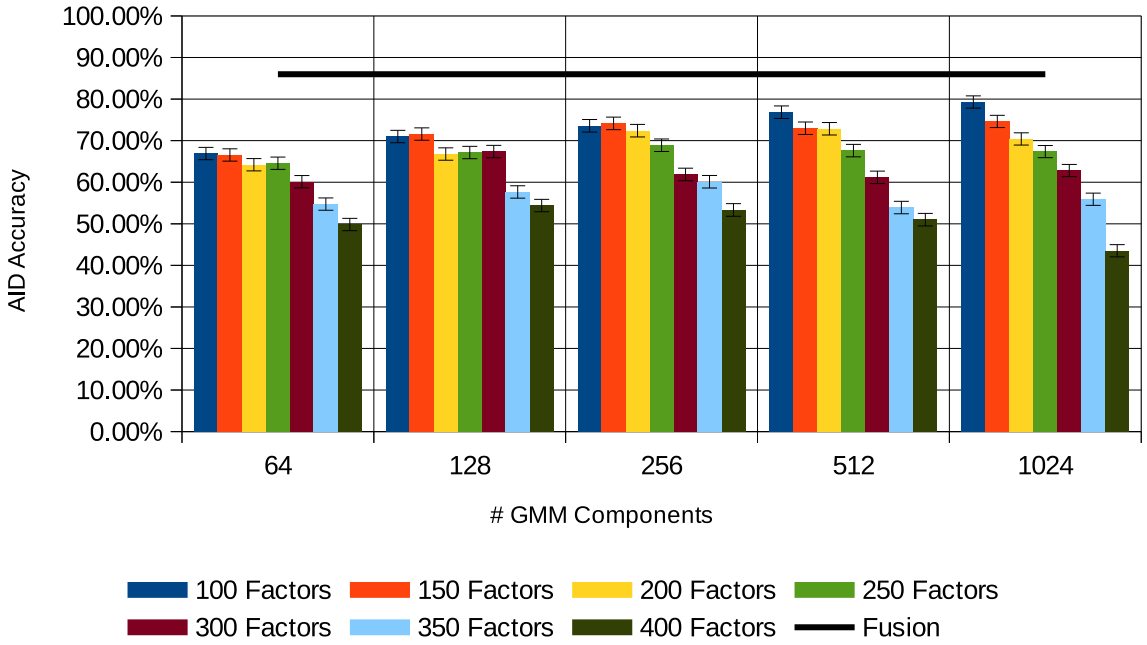


Figure 7.22: AID classification accuracy with fusion based on GA solution selection for LDA dimensionality reduction.

Of particular interest is the final choice of classifiers combined to form a majority vote classifier that achieves the optimised AID accuracy. This is shown in Table 7.5. For most factor dimensions, the GA selected higher GMM components, which we have already observed as giving better performance than lower GMM orders at an individual level. However, the combination of these together in a majority vote gives a considerable jump in performance. We can conclude, therefore, that the individual classifiers are better and/or worse on different utterances, further showing how the learnt projections, though very useful, are incomplete due to their assumptions.

Before proceeding further we note that the classifier combination technique we used (majority voting) is a very simple form of classifier combination. Other, more advanced, and potentially better techniques exist. We consider that an individual investigation for this problem area would be more than worthwhile, especially as corpora become more extensive and varied to reflect more realistic world scenarios.

Table 7.5: Genetic Algorithm selection for best classifier combination under for length-normalized i-Vectors, LDA projection and LDA classification.

K	100 Factors	150 Factors	200 Factors	250 Factors	300 Factors	350 Factors	400 Factors
64							
128					✓	✓	
256		✓	✓	✓			
512	✓			✓		✓	✓
1024	✓	✓		✓	✓		✓

Another note is that the criteria to achieve fusion was selected for expediency in obtaining an optimistic upper bound on performance. The GA was optimised in each iteration to reach a quasi-maximum score for the particular test set. For real-world scenarios, the use of another development dataset would be required to avoid an over-fit solution. The scope of the experiment here is analytical, and meant to show the gap in performance from the actual classifiers built so far, to what is theoretically possible to achieve using the same classifiers, if a sophisticated fusion mechanism is designed for this purpose. In utilising the test set itself for expediency, we have explicitly induced a substantial optimistic bias. The reader is referred to [211] for a discussion of over-fitting in model selection and bias in evaluation.

7.10 Alternative Projection Methods

The experiments carried out so far have given some insight into how the performance of classification with an i-Vector system has a strong dependency on at least three factors: the i-Vector configuration itself (GMM components, factor dimensionality), the supervised projection to suppress non-class information (LDA, QDA), and the classifier used on the dimensionality-reduced i-Vectors. We proceed in our experimentation with a number of alternatives to LDA and QDA projections. We propose a boosted i-Vector classification system (first described in [10]) that makes use of different projection methods to extract more class specific information than LDA alone.

7.10.1 Regularized linear discriminant analysis

In the case of high dimensionality feature vectors, LDA suffers from the small sample size problem, and has shortcomings such as the assumption of a common covariance matrix for all classes. There is no reason to consider that all accent classes satisfy the latter criterion. One

way of circumventing this assumption is to assume a separate covariance matrix for each class, leading to quadratic discriminant analysis (QDA). There is, however, an intermediate method between LDA and QDA, proposed by Friedman [212], termed regularized-LDA (R-LDA). In R-LDA, a regularization term is used to shrink the separate class covariance matrices in QDA towards a common covariance as in LDA.

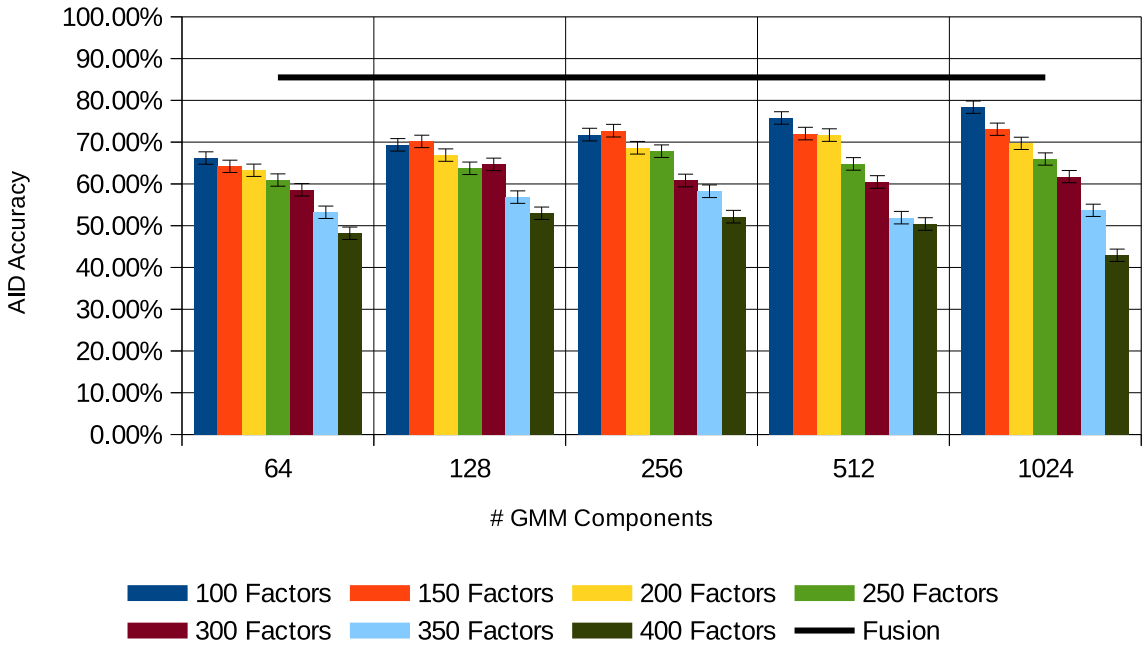


Figure 7.23: AID classification accuracy with fusion based on GA solution selection for regularized LDA dimensionality reduction.

Table 7.6: Genetic Algorithm selection for best classifier combination under for length-normalized i-Vectors, regularized LDA projection and LDA classification.

K	100 Factors	150 Factors	200 Factors	250 Factors	300 Factors	350 Factors	400 Factors
64							
128		✓	✓				
256	✓	✓	✓	✓		✓	✓
512	✓	✓	✓		✓		✓
1024	✓	✓		✓	✓	✓	✓

Figure 7.23 shows the results when LDA is replaced by regularized LDA. Performance is quite similar to standard LDA, with a slighlty lower fusion result of 85%. The best performing single classifier for this set performs at 78% accuracy, which is slightly lower than in LDA. The worst performing single classifier for this set performs at 43%, which is equivalent to standard LDA.

Looking at the final choice of classifiers combined to form a majority vote classifier that

achieves the optimised AID accuracy, shown in Table 7.7, there is a reliance on mostly higher component GMMs as expected, with some added dependence on lower GMM orders for 150 and 200 factor dimensions.

7.10.2 Semi-supervised discriminant analysis

Semi-supervised discriminant analysis (SDA) was proposed by Cai et. al. [213]. Similarly to RLDA, it also aims to overcome some of the problems with LDA, specifically that of not having enough training samples, and therefore creating an ill-formed projection. The idea of SDA is to use labelled data just like in LDA to maximize class separability, but also to use unlabelled samples to estimate the intrinsic geometric structure of the data. SDA is designed to estimate a projection that satisfies the LDA objective, but also avoids an 'overfit' in the data projection manifold. This is a very interesting idea for accent classification, since the different speakers in the three test sets can produce very different LDA projections. By using unlabelled test-set points at testing time, we build a smoother manifold, which is more representative of our test data.

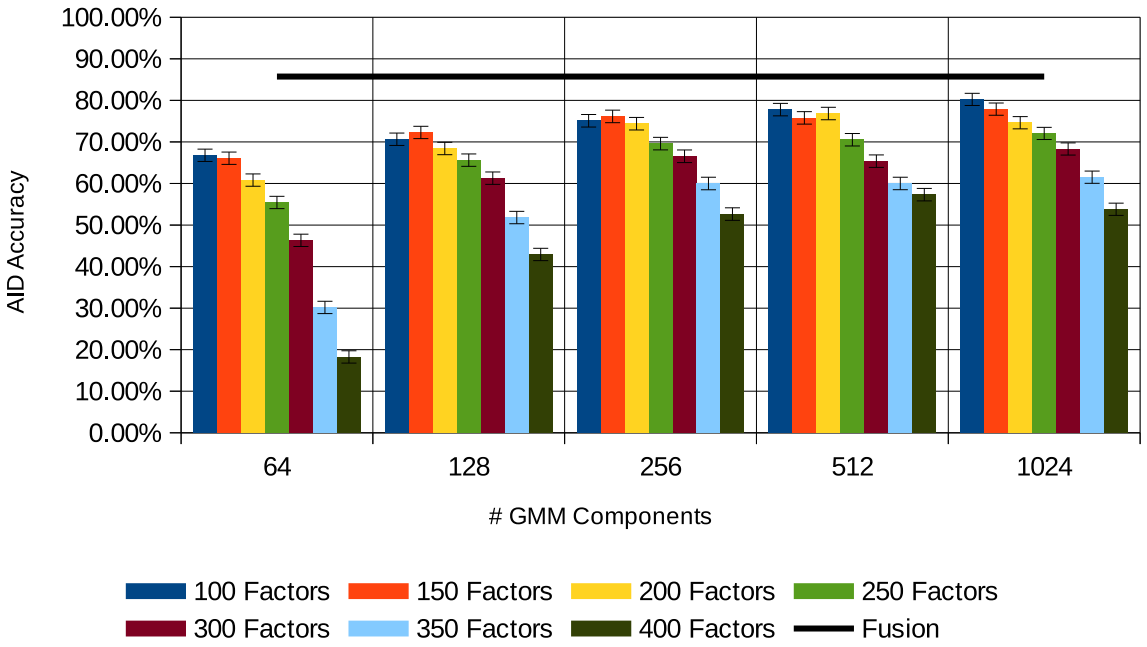


Figure 7.24: AID classification accuracy with fusion based on GA solution selection for SDA dimensionality reduction.

When LDA is replaced by SDA for dimensionality reduction, some interesting results emerge, as shown in Figure 7.24. Firstly, performance for all factor dimensions perform better on large order GMMs than in LDA. On low order GMMs however, performance in LDA is better. When

Table 7.7: Genetic Algorithm selection for best classifier combination under for length-normalized i-Vectors, SDA projection and LDA classification.

K	100 Factors	150 Factors	200 Factors	250 Factors	300 Factors	350 Factors	400 Factors
64							
128							
256	✓	✓	✓	✓			✓
512	✓		✓	✓	✓		✓
1024	✓	✓		✓	✓		

the same GA-based fusion optimization is performed to find a quasi-optimal combination of classifiers for a majority voting result, we obtain equal performance at 86%. The best performing single classifier for this set performs at 80% accuracy, which is slightly better than in LDA. The worst performing single classifier for this set performs at 18%, which is much worse than for the LDA testing set at 43%. The overall performance during fusion is however about the same.

The classifiers that were combined to form a majority vote classifier that achieves the optimised AID accuracy are shown in Table 7.7. Similarly to Table 7.5, for most factor dimensions, the GA selected higher GMM components, which we have already observed as giving better performance than lower GMM orders at an individual level, even for SDA dimensionality reduction.

7.10.3 Neighbourhood component analysis

Neighborhood component analysis (NCA) was proposed by Goldberger et. al. [214]. The technique is not part of the family of DA techniques, but is also a popular dimensionality reduction technique, and in various results such as [181], provides better language recognition results when compared to LDA. Unlike typical DA methods, NCA makes no assumptions about the shape of class distributions and the boundaries between them. It tries to utilize the power of k-nearest neighbour (KNN) classification for non-linear boundaries. In contrast to KNN, NCA is designed to learn a distance metric based on the labelled training data, since standard metrics such as Euclidean distance may be ineffective for problems such as language or accent classification. The projection and metric given by NCA minimizes the training error defined using leave-one-out cross validation, and is optimized so that 1-NN classification performs well afterwards. Since the final projection given by NCA is meant to perform well on 1-NN, the actual i-Vectors do not cluster into explicit accent groups as with discriminant analysis based projections. This can be seen in Figure 7.25.

Table 7.8: Genetic Algorithm selection for best classifier combination for length-normalized i-Vectors, NCA projection and 1-NN classification.

K	100 Factors	150 Factors	200 Factors	250 Factors	300 Factors	350 Factors	400 Factors
64	✓						
128	✓						
256	✓	✓	✓		✓		
512	✓	✓	✓	✓		✓	
1024	✓	✓	✓	✓		✓	

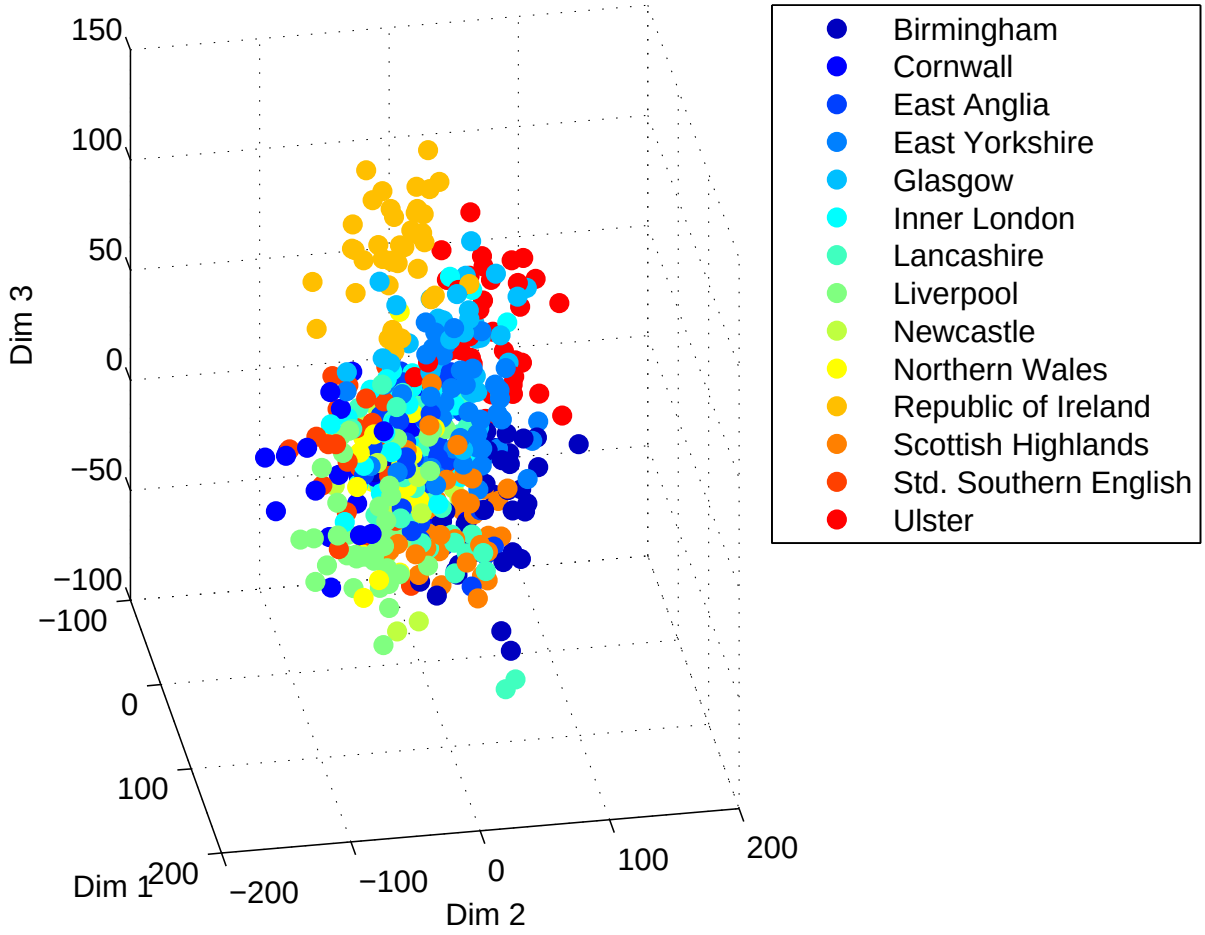


Figure 7.25: Projection of training data produced by NCA. Although some clustering is visible, it is not clear to the extent visible in projections based on discriminant analysis. The first three dimensions of the data obtained by NCA reduction are shown.

When LDA is replaced by NCA for dimensionality reduction, some interesting results emerge, as shown in Figure 7.26. Firstly, performance is much worse than for SDA and LDA dimensionality reduction for all GMM orders and factor dimensions. Higher GMM orders get the best performance for this test set. But the range of performance is very low between 6% (below chance level) to 50%. When the same GA-based fusion optimization is performed to find a quasi-optimal combination of classifiers for a majority voting result, we obtain equal

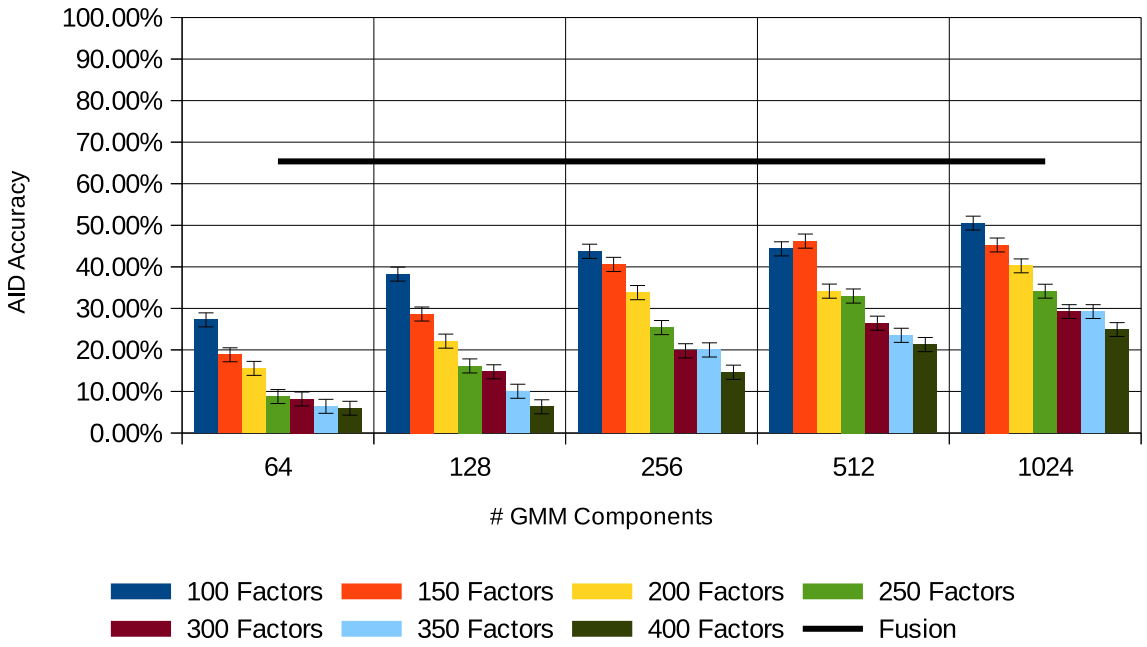


Figure 7.26: AID classification accuracy with fusion based on GA solution selection for NCA dimensionality reduction.

performance at 65%, which is of course worse than the SDA and LDA counterparts. A gain of around 15% from the best single classifier to the fusion system makes this projection the one that gains most fromm fusion given the poor initial performance of the single classifiers.

The classifiers that were combined to form a majority vote classifier that achieves the optimised AID accuracy are shown in Table 7.8. Similarly to Table 7.5, for most factor dimensions, the GA selected higher GMM components, which we have already observed as giving better performance than lower GMM orders at an individual level, even for NCA dimensionality reduction.

7.10.4 Combined Projection Fusion

The final complete results are obtained by a fusion of all projection methods for all GMM orders and factor dimensions. The results for all the approaches discussed so far, including the fused variants, as well as a combined fusion classifier are shown in Figure 7.27. For the combined projection fusion, a genetic algorithm was used to find a quasi-optimal set of classifiers to use for a majority voting result. In this case, there are 140 individual classifiers fed into the GA. The initial population is of 10,000 individuals, with a generation gap of 0.9, a crossover rate of 0.5, and a mutation rate of 0.0175. The GA runs for 200 generations.

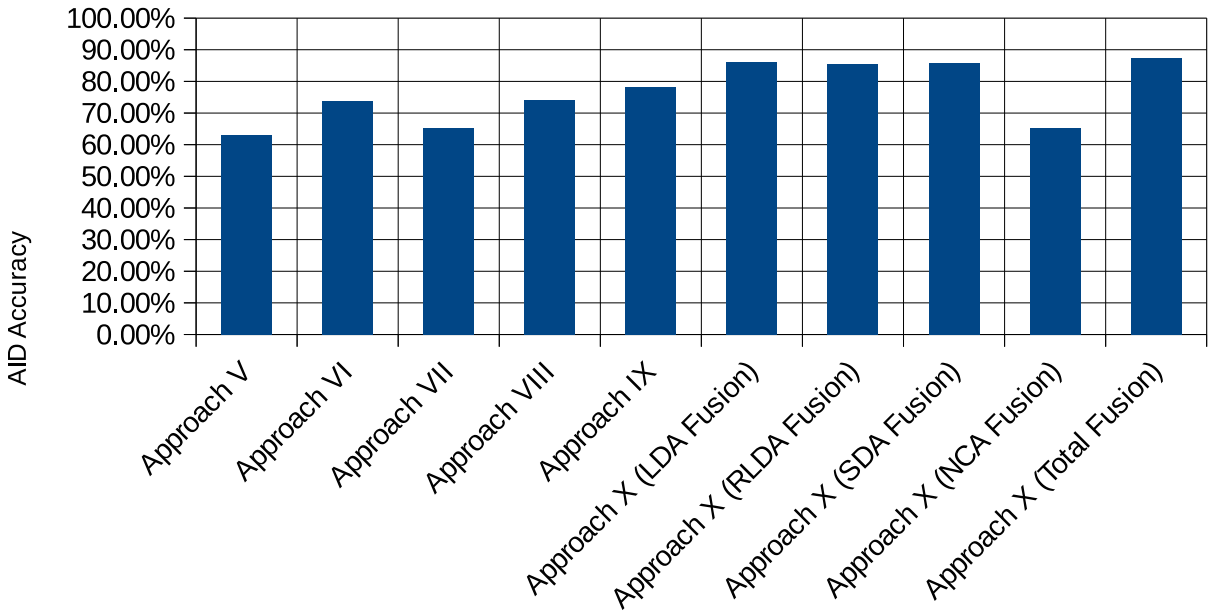


Figure 7.27: AID classification accuracy for all individual fusion systems compared with previous approaches, as well as a complete fusion system (best performance at 88%).

The final choice of classifiers combined to form a majority vote classifier that achieves the optimised AID accuracy is shown in Table 7.9. Many of the classifiers selected are those for larger GMM orders, which of course performed well at an individual level when compared to others. The classifiers from NCA projections, though giving poor performance when compared with the rest have also been utilized. In total 48 classifiers are selected by the GA solution finder.

7.11 Accent Confusion Analysis (Part 2)

We shall now have a second look at classification confusion and evaluate the benefits of our fusion system. A confusion matrix showing the correct versus predicted classification of utterances of the corpus for Approach X is shown in Table 7.12. A list of closest confusions per accent is given in Table 7.10. The results obtained are much improved over those obtained in Section 6.6, with an improvement over those obtained in Section 7.9 as well. Also, the results have maintained an overall association in the way accents are classified when comparing the classifications to the broad accent geographic formation (see Section 4.2), though a few confusions have been moved around from Section 7.9. Overall the amount of confused accents for each accent has been reduced considerably.

The Birmingham accent (North, Midlands) has been confused with the Inner London (South, London) and Newcastle (far North). The Cornwall accent (South, South West) has been confused

Table 7.9: The final choice of classifiers combined to form a majority vote classifier that achieves the optimised AID accuracy.

Method	K	100 Factors	150 Factors	200 Factors	250 Factors	300 Factors	350 Factors	400 Factors
LDA	64							
	128					✓	✓	✓
	256		✓		✓			
	512	✓			✓	✓		✓
	1024	✓	✓		✓	✓		
RLDA	64				✓			✓
	128				✓			
	256	✓	✓		✓			
	512					✓		✓
	1024	✓	✓		✓	✓	✓	
SDA	64							
	128							
	256	✓		✓	✓			✓
	512				✓			
	1024	✓	✓			✓		✓
NCA	64		✓	✓			✓	✓
	128		✓				✓	
	256	✓					✓	✓
	512	✓						
	1024	✓	✓	✓				

Table 7.10: Ordered list of closest confusions per each accent as given by the Approach X classifier. Where no confusions are made, columns are left empty.

Accent	Closest Accents												
brm	ilo	ncl											
crn	sse	nwa	brm	ean	eyk	ilo							
ean	brm	crn	sse	eyk									
eyk	lan	brm	nwa	ilo	sse								
gla	ncl	shl											
ilo	ean	sse	brm	crn	eyk								
lan	brm	eyk	nwa	gla	sse								
lvp	nwa	brm	ncl										
ncl	lvp	lan	nwa										
nwa	brm	crn	gla	ncl	sse	uls							
roi	brm	uls	crn										
shl	gla												
sse	brm	nwa	eyk	ean	crn								
uls													

with the Standard Southern English accent followed by Northern Wales and Birmingham. The Cornwall accent tends to be particularly confusing to the classifiers. The East Anglian accent is confused with Southern/Midlands accents such as Birmingham, Cornwall and Standard Southern English, as well as East Yorkshire. The East Yorkshire (North, Mid-North) accent is confused with the Lancashire (another North, Mid-North accent) and Birmingham (North, Midlands) accents. Glasgow (Scotland region) is only confused with the Newcastle (North, Far-North) and Scottish Highlands (Scotland region) accents. The Inner London accent (South, London), has a number of mixed confusions with accents of a few different regions. Scottish Highlands (Scotland region) is only confused with the Glasgow accent (the other Scottish region). The Standard Southern English accent is surprisingly one of the most confused accents, with confusions made for Birmingham, North Wales, East Yorkshire accents (all mid to far North accents), and East Anglia and Cornwall accents (the only southern accents). Aside from a few anomalies however, most of the confusions are related to broad accent and geographical location.

7.12 Leave-One-Speaker-Out (LOSO) Training

In order to test the effect of using more training data on classifier performance, we also performed tests where the classifiers were built using all the data available, except for that from a single speaker (leave-one-speaker-out training). The speaker who had been removed from the training data was tested, and results were pooled. Our comparison is based on LDA projection followed by LDA classification, as well as a local fusion based on these classifiers. The results are shown in Figure 7.28.

Table 7.11: Genetic Algorithm selection for best classifier combination for length-normalized i-Vectors, LDA projection and LDA classification, under LOSO training conditions.

K	100 Factors	150 Factors	200 Factors	250 Factors	300 Factors	350 Factors	400 Factors
64				✓			
128				✓			
256		✓		✓			✓
512		✓		✓			✓
1024	✓	✓			✓	✓	✓

The primary result is that the GA-optimized classifier selection results in a fusion performance of 89% accuracy, which is an increase of 3% over the fusion without LOSO training. This kind of improvement was expected. A more important result, however, is that all individual classifiers

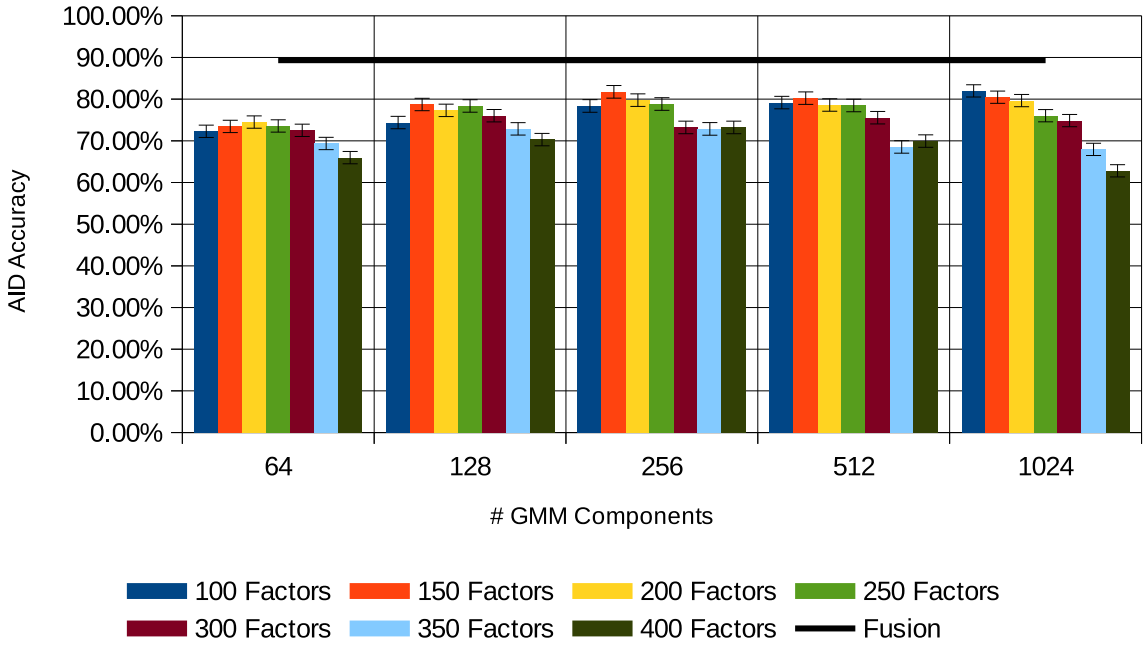


Figure 7.28: AID classification accuracy with fusion based on GA solution selection for LDA dimensionality reduction under LOSO training conditions.

perform reasonably well, and the range of accuracies lies between 63% to 82% i.e. a range of 19%. This is much less than the range observed without LOSO training, which stood at 36%. The trend therefore, is that with more training data, the difference in classifiers is possibly minimized. Another surprising result is that the best single classifiers for different factor dimensions do not necessarily occur at the higher order GMMs.

The final choice of classifiers combined to form a majority vote classifier that achieves the optimised AID accuracy is shown in Table 7.11. Though there is a selection from most of the classifiers based on the large 1024 component GMM, there is also a strong shift towards not selecting many classifiers or none at all from some factor dimensions (100/200/300/350). There is also a selection of classifiers with low GMM order for the case of 250 factor dimensions. The apparent availability of more data for training seems to put what have been gauged as 'weaker classifiers' in contention again. The results obtained here suggest that the optimum model parameter values seem to depend on the quantity of training data available, and no particular configuration should be ignored.

7.13 Summary

In this chapter we have applied the i-Vector paradigm for the problem of accent identification. We have shown a number of different approaches on how this may be achieved, getting different results from the same i-Vectors. We have introduced two new approaches to traditional classification systems. One system is based on an iterative "zooming-in" on classes in contention for a classification. Weak classes are removed at each iteration and as a result, the projection provided by discriminant analysis is optimized. This gives better results than standard classification methods. We have also developed a system of fused classifiers that together perform better than single classifiers. This shows that different i-Vector configurations learn different 'aspects' of the accent classes, and that the accuracy given by an LDA projection for a given configuration can be increased by multiple projections and multiple projection methods based on different i-Vector configurations. Finally, we have also examined the effect of more training data on the classification results, and showed how more data tends to shift the somewhat consistent trends observed in all previous experiments. In the next chapter, we shall have a look at how our accent identification method has been utilized within an automatic speech recognition system, with a summary of the results obtained in collaboration with researchers at the University of Birmingham. Following that, we study the performance of our accent identification system on short utterances (shorter than 30 seconds) and evaluate variations in performance by revisiting the frontend extraction process to find a better configuration. We also discuss another enhancement to the classification method, for an additional boost in performance of the classification system.

Table 7.12: Confusion matrix (in %) of correct vs predicted classification of utterance for the 14 accents of the British Isles for Approach X. Average accent recognition accuracy is of 87.37%. The diagonal, which represents the correct (no confusion) results is in **bold**, whilst off-diagonals (confusion) equal or greater than 8% are marked in **red**.

	brm	crn	ean	eyk	gla	ilo	lan	lvp	ncl	nwa	roi	shl	sse	uls
brm	**96.67**	0.00	0.00	0.00	0.00	1.67	0.00	0.00	1.67	0.00	0.00	0.00	0.00	0.00
crn	1.67	**83.33**	1.67	1.67	0.00	1.67	0.00	0.00	0.00	3.33	0.00	0.00	6.67	0.00
ean	7.02	5.26	**78.95**	3.51	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	5.26	0.00
eyk	2.67	0.00	0.00	**84.00**	0.00	1.33	8.00	0.00	0.00	2.67	0.00	0.00	1.33	0.00
gla	0.00	0.00	0.00	0.00	**96.67**	0.00	0.00	0.00	1.67	0.00	0.00	1.67	0.00	0.00
ilo	3.17	3.17	4.76	1.59	0.00	**82.54**	0.00	0.00	0.00	0.00	0.00	0.00	4.76	0.00
lan	**9.52**	0.00	0.00	**9.52**	1.59	0.00	**73.02**	0.00	0.00	4.76	0.00	0.00	1.59	0.00
lvp	1.67	0.00	0.00	0.00	0.00	0.00	0.00	**90.00**	1.67	6.67	0.00	0.00	0.00	0.00
ncl	0.00	0.00	0.00	0.00	0.00	0.00	1.67	5.00	**91.67**	1.67	0.00	0.00	0.00	0.00
nwa	4.76	4.76	0.00	0.00	1.59	0.00	0.00	0.00	1.59	**84.13**	0.00	0.00	1.59	1.59
roi	3.33	1.67	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	**91.67**	0.00	0.00	3.33
shl	0.00	0.00	0.00	0.00	1.52	0.00	0.00	0.00	0.00	0.00	0.00	**98.48**	0.00	0.00
sse	**10.42**	2.08	4.17	6.25	0.00	0.00	0.00	0.00	0.00	**8.33**	0.00	0.00	**68.75**	0.00
uls	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	**100.00**

Chapter 8

Short Utterance Classification, Frontend Parameters and AID in ASR

The previous chapter gave a very detailed development of accent identification (AID) under the i-Vector paradigm. The performance obtained increased steadily from methods prior to the i-Vector systems, and the fusion techniques we developed boosted AID performance considerably. To our knowledge, the performance obtained thus far is the best available for acoustic AID methods on the ABI-1 corpus. The combination of i-Vector modelling and projection methods like LDA manage to suppress the within-class speaker variation to a considerable level, and as a result the performance gain for AID is quite significant, and is bigger than gains observed for other problems like SID and LID.

However, there are a number of issues we have not yet addressed in this thesis. So far we have evaluated the i-Vector based AID system for test utterances that are 30 seconds long. In this chapter we will evaluate the performance of our system when test utterances are shorter in duration. The i-Vector relies on statistical information from frames of an utterance, and we anticipate degraded performance when using shorter training and testing utterances. We would like to estimate the degree to which this occurs. Also, we have so far relied on a single frontend extraction method throughout our experiments. Now that we have constructed a reasonably good AID classifier, this chapter will test a number of frontend extraction parameters to evaluate their effect on AID performance. As a consequence of the results observed here, we also propose further enhancements and obtain further AID classification improvements.

The chapter concluded by looking at the effect of such an AID system for the purposes

of ASR. This part of the work was done in collaboration with colleagues at the University of Birmingham. The chapter will give a short overview of the results, and a discussion on the implications of the results.

8.1 Short Utterance Classification

The nature of the i-Vector system relies in gathering time-invariant sufficient statistics over an utterance, and to use this information to obtain a point estimate in a total variability subspace. The subspace itself is estimated by the same statistics during training. Previous i-Vector work has shown how a limit on the available utterance length (and therefore, frame statistics for the utterance) can be a problem for making reliable estimates. Conversely there have also been cases where additional utterance length gives no further improvement, and chunking utterances for more i-Vectors for an utterance gives better modelling. The utterances we have used so far are the longest possible for the ABI-1 corpus i.e. roughly 30 seconds of speech. We therefore evaluate the performance of the system based on two shorter durations: ten seconds and three seconds. Therefore, each 30 second utterance is chunked into three utterances of ten seconds each, or into ten utterances of three seconds each. The process for training or testing the i-Vector system does not change, except that there are more individual i-Vectors for training and testing.

The first test performed is based on classifying length-normalized i-Vectors, projected via LDA, and classified with non-iterative LDA. The results for this test for 10 second utterances are shown in Figure 8.1. There are a number of points to discuss here. The first is the overall performance from the fused system (fusion optimised by GA, same as for previous 30 second tests). The reduction of utterance length for training and testing i-Vectors from 30 seconds to 10 seconds results in a performance drop of roughly 11%, from the previous 86% to 75%. Therefore, the expectation of degraded performance for shorter utterances is obvious.

Another observation we can make is that as with previous testing, a larger number of components for the GMM used to extract sufficient statistics results in overall better performance. The best single classifier, prior to fusion, is obtained for 1024 components, with 68% AID accuracy. Similarly to previous results, the general trend is for lower factor dimensionality gives better performance, and for this test, peak accuracy for the best single classifier is for 150 factors. Except for cases of larger numbers of GMM components at 512 and 1024, we can observe more stable performance across classifiers for different factor dimensionality. This was not observed for 30 second utterances. The overall decline in performance across increasing factor dimensionality

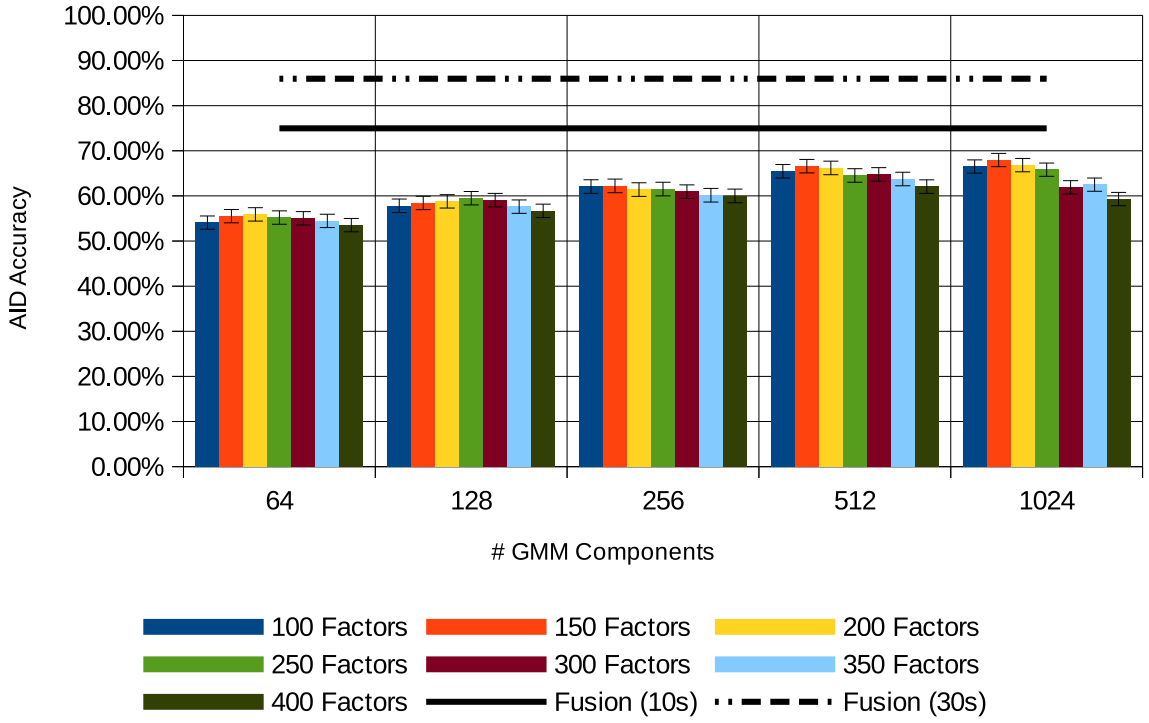


Figure 8.1: AID classification accuracy for i-Vectors that have been first length normalized and then projected to a lower dimensionality via LDA. Classification is performed via non-iterative LDA. Utterance duration is of 10 seconds. The previous result for the same test for 30 second utterance is shown for comparison.

for individual GMM configurations is not as steep as that observed for longer utterances.

A second test is performed, and this time the training and testing is based on utterances that are only three seconds long. The results of this test are shown in Figure 8.2. The overall performance from the fused system is further degraded with the very short three second utterances, with a drop of roughly 34%, from the previous 86%, down to 52% compared to the 30 second utterance tests. The performance drops from 75% down to 52%, a difference of roughly 23% when compared with the results for ten second utterances. It can be observed how the degradation in performance is non-linear with respect to utterance duration. Similar patterns of degradation are common, and we are also aware that beyond a certain utterance length, no further gain in performance may be obtained - this is certainly the case for speaker verification [215], where performance plateaus after a certain utterance duration, but degrades (somewhat non-linearly) with short utterances.

Even in this case, GMMs with a higher number of components give the best results overall, with the best single classifier prior to fusion obtaining 47% AID accuracy with 1024 components used for the GMM. Lower factor dimensionality also gives the best results, with 100 factors

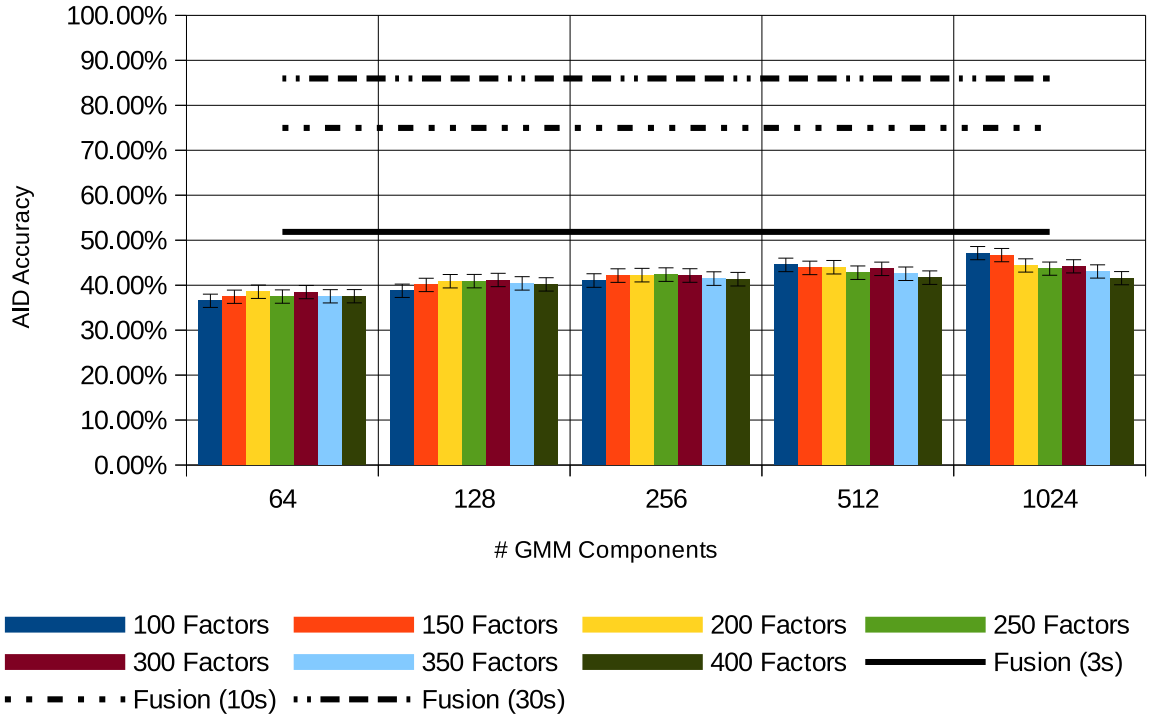


Figure 8.2: AID classification accuracy for i-Vectors that have been first length normalized and then projected to a lower dimensionality via LDA. Classification is performed via non-iterative LDA. Utterance duration is of 3 seconds. The previous results for the same test for 30 second and 10 second utterances are shown for comparison.

being used for this same classifier. It can also be observed that the difference in performance as factor dimensionality increases for a particular GMM configuration is not very steep. The general observation we gather here is that the factor dimensionality becomes increasingly more irrelevant as the utterance duration decreases. An additional observation is that mid-range factor dimensionality of 250 to 300 gives better performance in setups with a lower order GMM. These are different to the results obtained for longer utterance training and testing, and further highlight the need for appropriate parameter selection for the i-Vector systems.

If we look at gains observed over the best single classifiers by the fusion process, we see that for ten second utterances, the performance jumps from 68% to 75% (a jump of 7%). In the case of three second utterance, the performance jumps from 47% to 52% (a jump of 5%). In the case of 30 second utterances, the jump observed was of 7%, from 79% to 86%. It is interesting how for 30 and ten second utterances, the jump obtained by fusion is equivalent, and only degrades for three second utterances. Short duration utterances clearly remain a problem for the i-Vector domain, especially in the context of AID, which relies, perhaps even more than SID and LID, on differences in phonetic realisation of equivalent phrases or utterances.

8.2 Fronted Feature Extraction

The work presented in [153] evaluates AID under GMM-UBM classification. The features were modelled using inter-session variability compensation as described earlier in Section 3.6.3. The best AID performance for long utterances of 30s obtained was of roughly 60%. However, the work observed how different sub-bands are more useful than others for AID, with most information required for AID observed in the region of 0.34-3.44 kHz. This work motivated us to look into possible performance gains for different frontend configurations coupled with our very robust i-Vector based AID system.

We test three particular parameters of our MFCC extraction: the number of coefficients to extract, the number of filters in our filterbank, and the maximum bandwidth up to which the filterbank extends. So far, all the experiments performed to this point have kept these values constant, extracting 13 MFCCs over 30 filters that spread up to a bandwidth of 11.025 kHz, and we will take this system to be the reference to which we compare other combinations. The number of MFCCs extracted per frame was varied between 13 and 19. The number of filters used was varied between 20, 30 and 40. The maximum bandwidth up to which the filters extend was varied between 4 kHz, 8 kHz and 11.025 kHz. In total, there are 18 possible frontend extraction configuration.

Each extraction configuration is passed through the same process of training an i-Vector extractor for different i-Vector configurations (35 in total). We do not analyse the results of individual classifiers — this would obfuscate the results we really want from this experiment, that is the effect of frontend extraction parameters. Therefore, we only consider the fused results of each frontend system. The first result is shown in Figure 8.3, where the notation used e.g. 13/30/11025 refers to coefficients/filters/bandwidth respectively. The default configuration of 13/30/11025 performs reasonably well. However, there are multiple configurations that perform better. In particular, the group of results with a bandwidth of 8 kHz gives the best overall performance. Fusion from SDA performs better with a 13/20/8000 configuration, whilst fusion from LDA performs better with a 13/30/8000 configuration. Interesting, a particular system of 13/30/4000 from the 4 kHz group performs better than our default system.

The second result is shown in Figure 8.4. This is an equivalent experiment, except that it is performed for utterances of ten seconds duration. The default configuration of 13/30/11025 performs reasonably well. However, the group of results with a bandwidth of 8 kHz give superior performance overall. Roughly equivalent performance is observed for the 13/20/8000

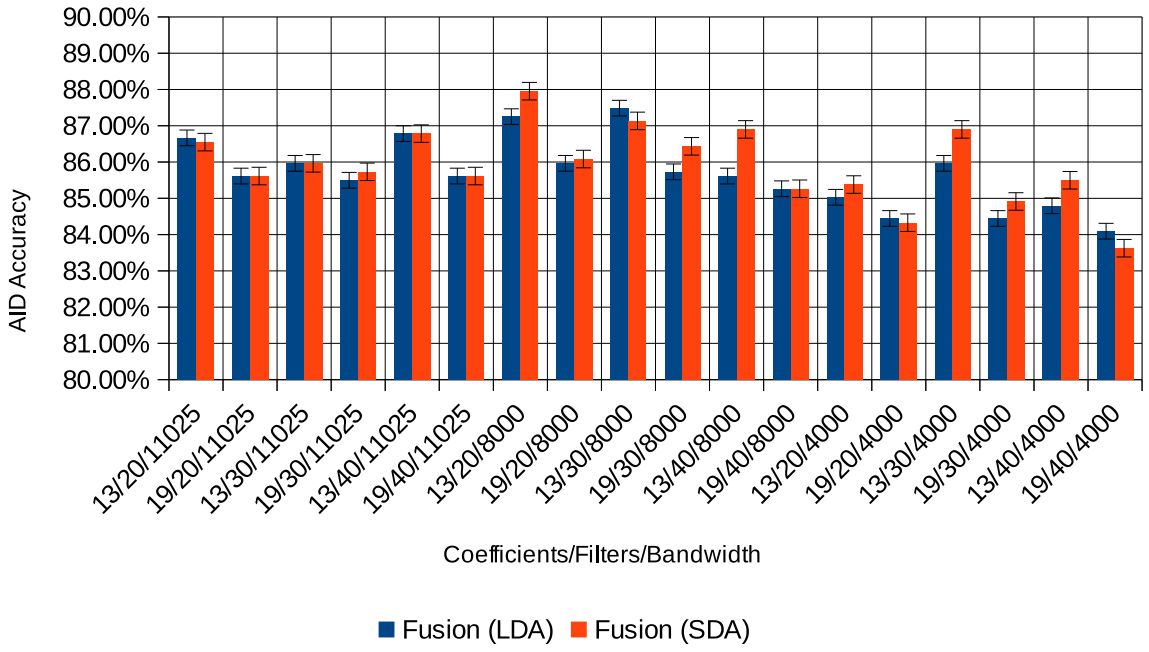


Figure 8.3: AID classification accuracy for i-Vectors that have been first length normalized and then projected to a lower dimensionality via a LDA and SDA projection. Classification is performed via non-iterative LDA. Utterance duration is of 30 seconds. The reference result configuration is marked 13/30/11025, which represents the default frontend configuration used so far in previous chapters.

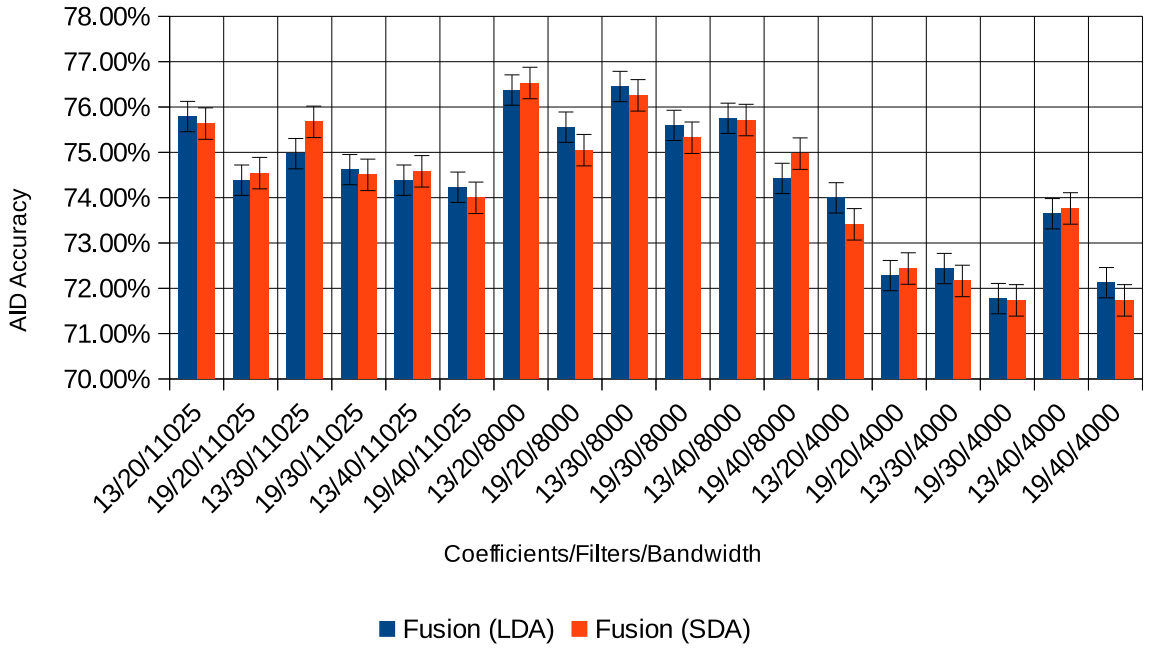


Figure 8.4: AID classification accuracy for i-Vectors that have been first length normalized and then projected to a lower dimensionality via a LDA and SDA projection. Classification is performed via non-iterative LDA. Utterance duration is of 10 seconds. The reference result configuration is marked 13/30/11025, which represents the default frontend configuration used so far in previous chapters.

and 13/30/8000 configurations. This corroborates the results observed with 30 second utterances. Conversely, the results for 4 kHz bandwidth are very poor for the ten second utterances.

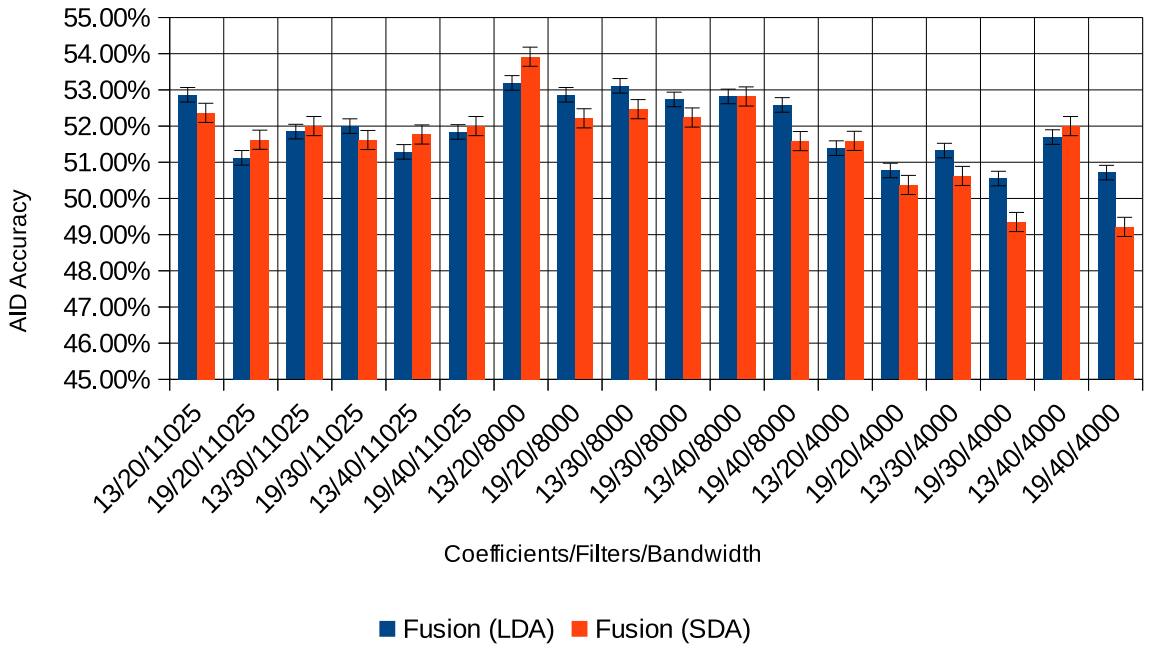


Figure 8.5: AID classification accuracy for i-Vectors that have been first length normalized and then projected to a lower dimensionality via a LDA and SDA projection. Classification is performed via non-iterative LDA. Utterance duration is of 3 seconds. The reference result configuration is marked 13/30/11025, which represents the default frontend configuration used so far in previous chapters.

The second result is shown in Figure 8.5. This is an equivalent experiment, except that it is performed for utterances of three seconds duration. The default configuration of 13/30/11025 does not perform badly. However, the observations made in previous tests also apply here, and the 8 kHz bandwidth group is the best of all three. The best result is obtained with the 13/20/8000 configuration, and since this occurs for the previous experiments on longer duration utterances, we take this to be the best global configuration for AID on the ABI-1 corpus. Again, the results for 4 kHz bandwidth are very poor when compared to other groups.

8.2.1 Multiple Frontend and Projection Fusion

Considering the fact that we have 18 different frontend configurations for a particular projection method, and that each configuration further subdivides into 35 different i-Vector configurations, a further test that we can perform is to fuse all the systems together, for a total of 630 'weak' classifiers. Further more, the 630 'weak classifiers' are available for every projection we utilize, of which there are four, resulting in a total of 2520 individual classifiers. We use the term 'weak'

very loosely here. Each of these systems is of course, a complete i-Vector system, which is not usually considered a 'weak' classifier. But within the context of using an ensemble of 2520 different systems, we can refer to each system as being a 'weak' learner.

We do not perform fusion on this entire block altogether. Instead, we perform fusion for every projection individually, and the heuristic solutions from each projection are combined together as is. To perform fusion of each projection, the genetic algorithm is setup with each "chromosome" as a binary vector of 630 entries, with each binary entry indicating whether a particular classifier output should be considered in the majority vote (1) or not (0). The initial population is of 1000 individuals, with a generation gap of 0.9, a crossover rate of 0.5, and a mutation rate of 0.0175. The GA runs for 200 generations.

The results (in confusion matrix form) for this fusion are presented at the end of this chapter in Tables 8.2, 8.3 and 8.4 for 30 second, 10 second and three second utterances respectively. The average AID accuracy obtained is 90.18%, 80.16% and 57.02% for 30 second, 10 second and 3 second utterance respectively. If we look at gains observed over the results in the previous section, in the case of 30 second utterances, the jump observed was of 2%, from 88% to 90% (when compared to results in Section 7.10.4). For ten second utterances, the performance jumps from 75% to 80% (a jump of 5%) and in the case of three second utterance, the performance jumps from 52% to 57% (a jump of 5%). A summary of performance of each individual accent based on different utterance lengths is presented in Figure 8.6. The accents are sorted in descending order of AID accuracy, from "easier" to classify to "harder" to classify accents according to 30 second tests.

It is interesting to observe that the accents of the Scottish Highlands/Glasgow/Ulster/Republic of Ireland (rhotic accents) and Liverpool/Cornwall (lightly rhotic accents) are the easier accents to classify, whereas all other accents, which are mostly non-rhotic are harder to classify (based on the rhotic/non-rhotic classification of English dialects in the 1950s [216]. It is also interesting to note that the order of best classified accents changes when the utterance length changes. The worst performing classification is for Standard Southern English on 30 second and 10 second utterances, whilst for three second utterances, Birmingham proves to be the hardest accent to identify. The best performing classification is for Scottish Highlands for all utterance durations. The Glasgow accent is equally identifiable at 100% for 30 second utterances.

There is a mix of broad geographical locations across Figure 8.6. It is interesting to note that Scottish and Irish accent regions are located in the first half of the chart, together with two other northern-mid-northern accents (Liverpool/Birmingham). Only one southern accent (Cornwall)

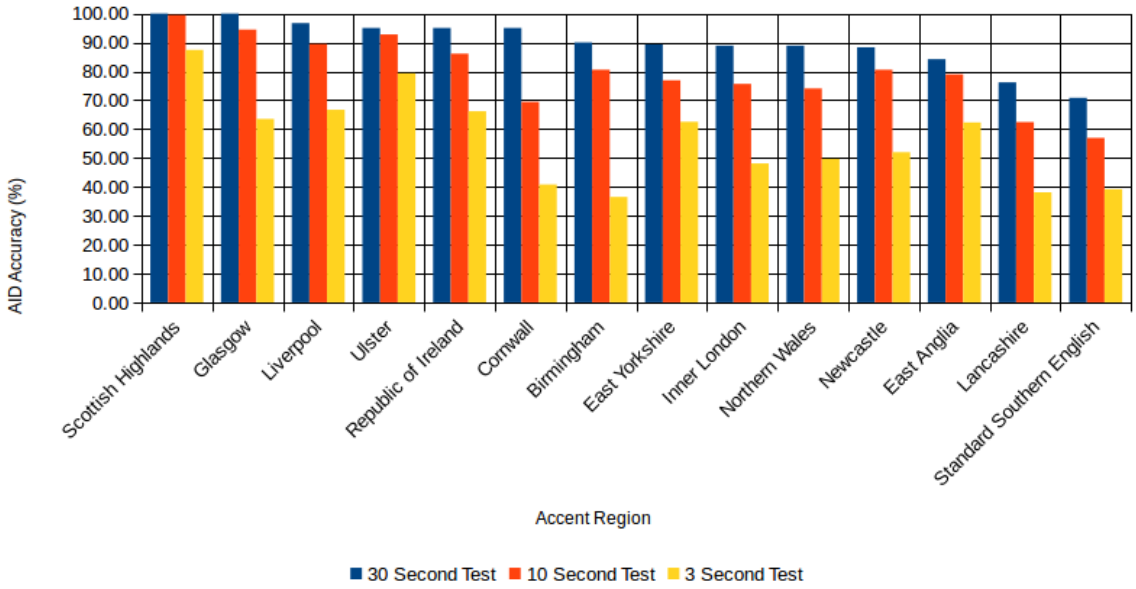


Figure 8.6: AID classification accuracy for individual accents, sorted by AID accuracy.

is in the first half of the chart. In the bottom half, there is a broader mix of accent regions. The majority of southern accents (Standard Southern English, East Anglia, Inner London) are in the bottom half of the chart. This suggests that southern accents are generally harder to distinguish. However we also find Wales, as well as two northern/mid-northern accents (East Yorkshire and Newcastle) in the bottom half.

Given the nature of fusion with multiple frontends, projection methods, and i-Vector parameters, this performance is only achieved with the creation of multiple i-Vector estimates for a given utterance. This tends to slow classification down when done in a linear fashion, and therefore, for more practical use, it is suggested that each feature extraction method, and consequently each i-Vector is estimated in some parallel processing architecture. Whilst this kind of processing is becoming common in speech processing nowadays, with the utilisation of graphical processing units, we do not do so in this thesis. However, we emphasize the point that such a frontend is not impractical.

8.3 AID for Speech Recognition

One of the problems in speech recognition is the influence accented speech has on recognition word error rate. An ASR system adapted to a particular speaker or accent group can, of course, mitigate this problem. However, within the context of deploying a speech recognition for use with

multiple speakers and multiple accents, it is not always possible to have a prior adapted model that is specific to a particular accent group. This problem has been studied before, and many different proposals have been made, including adapting pronunciation models [217, 218, 219], creating accent-independent models by incorporating training data from multiple accents [220], the incorporation of accent information in HMM decision tree clustering [220], adaptation of features [221], utilizing features that discriminate across accents [222], and careful selection of training data from varied material sources [223, 224].

In work [3] that came out of a collaboration with colleagues at the University of Birmingham, a new approach was proposed. The approach was to have specific ASR models tailored at recognising speech of individual accents. During recognition, an initial analysis is made from the first seconds of data, and AID is performed to determine the most likely accent, and therefore, the specific model to use for speech recognition. The work in this thesis is not concerned with the creation of accent-specific ASR models (read [3] for details of this), but on the utilization, and advantages of our i-Vector based AID system for this particular purpose, when compared to other, more traditional AID techniques.

Two AID techniques were tried. The first was a phonotactic AID system based on parallel phone recognition followed by language modelling (PPRLM). This requires phone recognition, vectorization and an SVM backend to learn and discriminate between accents. The second technique was the i-Vector AID system being developed in this thesis. At the time of the experiments in [3], the AID accuracy of these systems were at 19.30% and 18.95% respectively, with the i-Vector AID system having been presented earlier in [10]. The performance was therefore roughly equivalent on AID for the ABI-1 corpus.

Although we do not deal with the adaptation technique itself, it is important to understand the different kinds of adaptation performed to evaluate the importance and relevance of AID to ASR. Six different types of adaptation are performed, which we elicit below:

- Baseline (B0) performance — a speech recognition system is trained on the WSJCAM0 corpus. This ASR system is then used to perform recognition of one long (40 second) utterance of speech per speaker in the ABI corpus.
- SSE adaptation (B1) performance — the speech in the WSJCAM0 corpus can be considered close to the standard southern English (SSE) accent. In order to make sure that the ASR system we want to adapt is not influenced by task/corpus shift, the baseline is adapted with SSE data before prior testing recognition of the same long utterance of speech per

speaker. This gives an indication of dataset shift on recognition performance.

- AID-dependent model through "correct" accent knowledge (B2) — in this recognition test, it is assumed that we know which accent is associated with each tested utterance, and the ASR system adapted to that particular accent is used. This experiment gives a theoretical peak performance when the actual accent is known prior to recognition (an "oracle" system).

- Unsupervised speaker adaptation (S0) — is it better to perform speech recognition with an accent-adapted model for arbitrary speakers of that accent, or is it better to have speaker-specific ASR models with no explicit regard for general accent? In this experiment, recognition is performed on MLLR-adapted ASR models.

- AID-dependent model through i-Vector decision (A0) and phonotactic decision (A1) — in this recognition test, the accent for a test utterance is determined by the i-Vector or phonotactic based AID systems, and the ASR system adapted to that particular accent is used.

- AID-dependent model followed by unsupervised speaker adaptation (BS) and (AS) — for each speaker, model selection is performed using either "correct" accent (BS) or using phonotactic AID (AS). The selected accent model is then adapted further to the individual speaker with unsupervised MLLR adaptation.

Table 8.1: Comparison of results for all ASR experiments.

Experiment	Data from test speaker (seconds)	WER(%)
B0	—	26.0
B1	—	28.7
B2	—	14.7
S0	48.0	20.37
S0	101.5	18.75
S0	136.0	18.99
S0	221.0	17.83
A0	43.2	15.2
A1	43.2	15.3
BS	43.2	13.7
AS	43.2	14.1

A summary of results from these experiments is given in Table 8.1. A number of very interesting results emerge. Out of the baseline systems (B0, B1 and B2), B2 which used the "correct" accent model for recognition gives a word error rate (WER) of 14.7%. This is clearly much lower than results for B0 and B1, demonstrating that accent-specific modelling (as expected)

will give better performance in recognition. Despite a rough AID classification error of almost 20%, A0 and A1 give very close performance at 15.2% and 15.3% WER respectively. Performing unsupervised speaker adaptation (S0) gives varying degrees of performance depending on how much speaker data is used for adaptation, but even with maximal adaptation data of 221 seconds, performance is still worse than for AID selection methods. The combination of accent model selection followed by speaker adaptation (BS for "correct" model selection and AS for AID model selection) gives some slight improvement with error rates dropping to 13.7% and 14.1% respectively. The results show how the performance in AID systems is a good motivator to apply AID to utterances prior to accent-dependent model recognition, despite the inherent error of AID classifiers. It seems that wrongly selected accent models (due to AID errors) are still providing a selection that is "close" in accent space to the "correct" accent, and hence the word error rate does not increase much. Furthermore, AID can now be performed without requiring a seemingly more complicated AID system such as a phonotactic PPRLM system, but an acoustic approach is sufficient.

It is interesting to break down these results according to individual accents. An extract of these results by accent are shown in Figure 8.7, which is extracted directly from [3]. The results are ordered with the baseline system B0 as a reference from worst to best word error rate. By looking at the performance of baseline systems B0 and B1, it is easy to see that there is a wide range of WER depending on which accent is analysed. The best WER is of course located at the SSE accent, which is the closest accent group to the original WSJCAM0 corpus used for training the baseline ASR system. Performance degrades depending on how close or far the accent is to SSE. On the other hand, the range in WER is much narrower across all accents from the B2, A0, A1 and AS systems, which are all dependent on accent-specific models being selected. Though there is still a common trend of WER decreasing from left to right (suggesting more adaptation data is required, or that the particular accent is hard for recognition purposes), this range appears smooth with respect to what is observed in B1, B0 and S0.

The main requirement in this study is that a 43 second utterance (around 30 seconds of actual speech data) was required for model selection. This is of course quite long, and it would be ideal to obtain reliable model selection on short duration utterances. The results earlier on in this chapter focus on this aspect of short duration classification. Given that the ASR performance seems insensitive to AID accuracy with a 20% error margin, we can already presume that the results obtained for 10 second AID (with a similar 20% error margin) are already sufficient for this purpose, although this has not been evaluated experimentally yet. With further progress, it

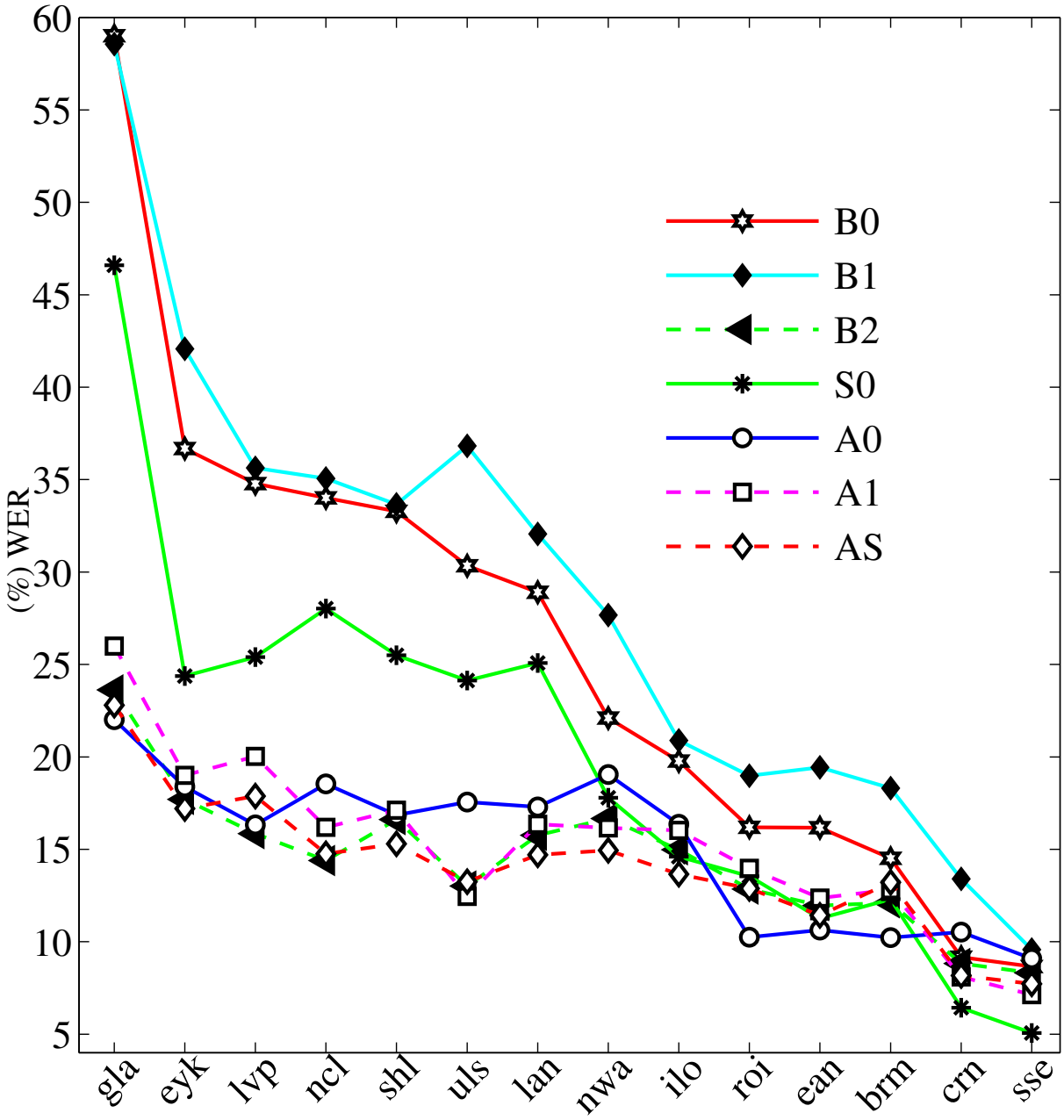


Figure 8.7: Comparison of ASR results by accent [3].

might even be possible to obtain a reliable model selection based on just a few seconds (e.g. three seconds) of data, which would be a good achievement for applying acoustic AID to ASR systems.

Whilst this approach was being tested, there was an issue on whether the comparisons being made are "fair" with respect to the different amounts of data available for speaker or accent-based adaptation. The offline accent-based ASR models had data running into around 5 hours for adaptation. On the other hand, speaker based adaptation had much less data, with a cap of around 220 seconds. However, the key focus of this study was to demonstrate that

reliable accent model selection can be performed in "real-time" with a small amount of data. The consequence is that major WER gains can be achieved with a 30 second cut by selecting a reliable ASR model. By demonstrating that the same AID performance available in [10, 3] is now possible with 10 second cuts, the argument of augmenting an acoustic AID classifier for ASR by model selection is strengthened even further. However, without experimentation, we do not know whether similar advantages can be obtained with three second cuts. The current performance of around 57% AID accuracy may be too low to be incorporated into an ASR system. However, we expect that with further boosting this result, the combination of acoustic AID methods for model selection in ASR becomes even more practical.

8.4 Summary

This chapter served to refine the work in this thesis. We investigated the effect of using different length utterances for training and testing on the performance of our i-Vector system. As expected, performance degrades with shorter utterances. We then looked at the effect of changing frontend filterbank parameters and assessed the effect on AID performance. It was noticed that there is considerable variation in performance from configuration to configuration. Also, when a multi frontend and multi-configuration i-Vector system was used to fuse many 'weak' classifiers together, performance was optimised further. The results presented are, as far as we can tell, the state-of-the-art for acoustic AID on the ABI-1 corpus. Finally, we looked at an initial study of utilising acoustic AID for model selection in ASR. The results shown here are promising, and with further advances in AID accuracy on short utterances, we can expect this proposal to be practical for ASR design. We found that the correct accent identity is not strictly required for model selection in ASR. An error margin of around 20% for a typical AID classifier constructed in this thesis had only a small effect on WER in the ASR experiments conducted in our collaborative experiments with the University of Birmingham. This suggests that using an accent that is "close" to the correct accent is good enough for model selection. This is advantageous, and we can presuppose the possibility of including a vector showing confidence in the closest (say two or three) accents for an utterance as part of the frontend to the ASR system, thereby incorporating the model selection as part of the feature frontend, rather than as a separate module. This may have practical implications for the deployment of large-scale ASR systems for multiple languages, where multiple accent models for each language may not be desirable.

This chapter concludes the work in this thesis, and the following chapter will conclude the thesis with a final discussion.

Table 8.2: Confusion matrix (in %) of correct vs predicted classification of utterance for the 14 accents of the British Isles for 30 second utterances with a fusion of 2520 classifiers. Average accent recognition accuracy is of 90.18%. The diagonal, which represents the correct (no confusion) results is in **bold**, whilst off-diagonals (confusion) equal or greater than 8% are marked in **red**.

	brm	crn	ean	eyk	gla	ilo	lan	lvp	ncl	nwa	roi	shl	sse	uls
brm	**90.00**	0.00	1.67	0.00	0.00	5.00	0.00	0.00	0.00	1.67	0.00	0.00	1.67	0.00
crn	0.00	**95.00**	0.00	0.00	0.00	1.67	0.00	0.00	0.00	0.00	0.00	0.00	3.33	0.00
ean	3.51	5.26	**84.21**	1.75	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	5.26	0.00
eyk	2.67	0.00	0.00	**89.33**	0.00	0.00	5.33	0.00	0.00	1.33	0.00	0.00	1.33	0.00
gla	0.00	0.00	0.00	0.00	**100.00**	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
ilo	6.35	1.59	0.00	3.17	0.00	**88.89**	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
lan	6.35	0.00	0.00	7.94	0.00	3.17	**76.19**	0.00	0.00	6.35	0.00	0.00	0.00	0.00
lvp	0.00	0.00	0.00	0.00	0.00	0.00	0.00	**96.67**	0.00	3.33	0.00	0.00	0.00	0.00
ncl	1.67	0.00	0.00	5.00	0.00	0.00	1.67	0.00	**88.33**	3.33	0.00	0.00	0.00	0.00
nwa	4.76	1.59	0.00	0.00	1.59	0.00	0.00	0.00	0.00	**88.89**	0.00	0.00	1.59	1.59
roi	3.33	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	**95.00**	0.00	0.00	1.67
shl	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	**100.00**	0.00	0.00
sse	**8.33**	2.08	4.17	**10.42**	0.00	0.00	0.00	0.00	0.00	4.17	0.00	0.00	**70.83**	0.00
uls	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	5.00	0.00	0.00	**95.00**

Table 8.3: Confusion matrix (in %) of correct vs predicted classification of utterance for the 14 accents of the British Isles for 10 second utterances with a fusion of 2520 classifiers. Average accent recognition accuracy is of 80.16%. The diagonal, which represents the correct (no confusion) results is in **bold**, whilst off-diagonals (confusion) equal or greater than 8% are marked in **red**.

	brm	crn	ean	eyk	gla	ilo	lan	lvp	ncl	nwa	roi	shl	sse	uls
brm	**80.56**	1.11	0.56	0.00	0.00	3.33	3.33	1.11	2.78	2.22	0.56	0.00	4.44	0.00
crn	0.56	**69.44**	0.00	2.22	1.67	**8.89**	0.00	0.00	2.78	3.89	0.00	1.11	**9.44**	0.00
ean	4.68	5.26	**78.95**	1.75	0.00	0.00	0.58	0.00	0.00	0.00	0.00	0.00	**8.77**	0.00
eyk	3.56	1.33	0.00	**76.89**	0.00	0.00	**8.89**	0.44	1.33	5.78	0.44	0.00	1.33	0.00
gla	0.56	0.56	0.00	0.00	**94.44**	0.00	0.00	0.56	0.00	0.56	1.11	1.67	0.00	0.56
ilo	**8.89**	5.82	2.12	5.29	0.53	**75.66**	0.53	0.00	0.00	0.53	0.00	0.00	0.53	0.00
lan	6.35	1.59	0.00	**13.76**	3.17	1.59	**62.43**	0.00	0.53	7.94	0.00	0.53	2.12	0.00
lvp	0.56	2.22	1.11	0.56	0.56	0.00	0.00	**89.44**	0.56	4.44	0.00	0.56	0.00	0.00
ncl	2.22	1.11	0.00	3.89	0.00	0.56	2.78	2.78	**80.56**	4.44	0.00	0.00	1.67	0.00
nwa	5.82	5.29	0.00	0.00	0.53	1.06	0.53	1.06	4.23	**74.07**	1.59	0.00	4.23	1.59
roi	2.22	0.56	0.00	0.00	0.00	0.00	0.00	0.56	1.67	0.56	**86.11**	0.00	0.56	7.78
shl	0.00	0.00	0.00	0.00	0.51	0.00	0.00	0.00	0.00	0.00	0.00	**99.49**	0.00	0.00
sse	**11.81**	**9.03**	3.47	**8.33**	0.00	2.08	0.00	0.00	3.47	4.86	0.00	0.00	**56.94**	0.00
uls	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	7.22	0.00	0.00	**92.78**

Table 8.4: Confusion matrix (in %) of correct vs predicted classification of utterance for the 14 accents of the British Isles for 3 second utterances with a fusion of 2520 classifiers. Average accent recognition accuracy is of 57.02%. The diagonal, which represents the correct (no confusion) results is in **bold**, whilst off-diagonals (confusion) equal or greater than 8% are marked in **red**.

	brm	crn	ean	eyk	gla	ilo	lan	lvp	ncl	nwa	roi	shl	sse	uls
brm	**36.50**	7.17	2.33	1.83	1.67	**9.50**	5.67	2.83	3.00	**13.67**	3.00	0.17	**10.50**	2.17
crn	1.67	**40.83**	2.00	2.67	3.17	**8.83**	1.50	5.33	**10.67**	**9.17**	2.17	1.83	**8.83**	1.33
ean	7.72	4.39	**62.28**	7.02	0.53	1.23	4.39	0.53	0.35	0.18	0.53	2.28	**8.42**	0.18
eyk	4.00	2.13	2.53	**62.53**	1.07	1.87	**10.00**	1.07	1.87	4.13	0.40	5.87	2.13	0.40
gla	2.33	2.83	0.67	1.17	**63.50**	2.67	2.33	**13.00**	2.33	3.67	1.50	2.33	1.17	0.50
ilo	7.62	**10.32**	3.02	7.62	3.33	**48.10**	5.24	2.54	3.81	3.49	1.27	0.16	2.86	0.63
lan	**9.68**	1.75	3.02	**15.56**	4.29	**8.25**	38.10	1.43	1.59	**8.73**	0.95	0.79	4.92	0.95
lvp	2.67	6.17	0.33	1.17	2.00	3.33	1.00	**66.67**	6.33	**9.50**	0.50	0.17	0.00	0.17
ncl	3.33	7.00	1.17	2.67	3.17	1.83	3.50	**8.00**	**52.00**	**9.33**	1.67	1.83	2.83	1.67
nwa	6.19	**9.05**	0.00	0.79	2.06	3.65	2.06	3.17	**9.21**	**49.68**	2.06	0.63	7.94	3.49
roi	4.67	2.67	0.33	0.33	1.33	1.67	0.17	1.17	2.83	1.83	**66.17**	0.50	1.67	**14.67**
shl	0.15	2.27	0.61	3.79	1.36	0.76	0.15	1.36	0.45	1.06	0.00	**87.42**	0.00	0.61
sse	**10.21**	**8.96**	4.17	7.29	1.25	6.67	2.50	0.63	7.08	**8.96**	1.67	0.21	**39.17**	1.25
uls	0.67	0.83	0.17	0.00	1.00	0.33	1.50	0.00	1.33	1.33	**12.33**	0.50	0.67	**79.33**

Chapter 9

Conclusion

This thesis has presented novel research in the field of automatic gender and accent identification using acoustic methods. The premise of identification from speech relies on mapping acoustic correlates from speech to characteristic categories such as speaker, language, gender and accent. The thesis began by proposing more robust techniques for gender classification, but the bulk of it has focused on acoustic methods for accent identification. Starting from first principles, data-driven experimentation guided the work, resulting in a highly optimised AID system, giving state-of-the-art performance on the ABI-1 corpus. The main focus was primarily on the application of standard methods to AID, in order to identify baseline performance and assess the difficulties of the problem. The thesis then made use of the i-Vector paradigm for AID, identifying the fact that performance, though reasonable, is not on par with other applications of the i-Vector paradigm in SID, LID etc. For this reason, a comprehensive investigation was made by assessing dimensionality reduction projections, i-Vector configuration parameters, frontend features, as well as classification techniques. The performance in AID varies considerably depending on the specific configuration from the feature extraction stage, all the way up to the classification methodology, and by the end of the thesis, we consider a system made up by fusing the outputs of many 'weak' classifiers. The term 'weak' should be considered loosely here, as each of these 'weak' classifiers is usually sufficient for other problems such as SID and LID. We feel this thesis provides sufficient evidence that this is not the case for AID. Furthermore, with colleagues at the University of Birmingham, we experimented with the use of acoustic methods of AID for the purpose of ASR applications, demonstrating practicable use given the optimised performance obtained in this research.

This chapter concludes the thesis. We first given an overview of each chapter. Following

this, we provide a broad discussion on the achievements and results. We then give some insight into possible future work arising from this thesis, and mention some important requirements which we feel are necessary to develop this field further.

9.1 Thesis Overview

The introduction to this thesis (Chapter 1) introduces the field of category identification from speech, which is described as a signal which conveys multiple layers of information to the listener. Given that the thesis focuses mainly on accent identification, we define the idea of accents as understood in this thesis. Mention is made of the state of research in this field, the increasing interest in the problem, and a brief mention of what seems to be problematic in AID — speaker variability. Much of the work in this thesis focuses on discovering the extent to which this problem can be mitigated within the i-Vector paradigm. Specific research questions are listed, and we feel this thesis has provided some degree of an answer to each.

Chapter 2 provides a background to theory in speech processing and machine learning that is essential to understand and work in the different aspects of this thesis. An overview of speech production in humans is given, and how speech can be processed using a frontend system to produce the feature vectors required for the work in this thesis. Furthermore, we demonstrate a number of classical feature modelling techniques which are at the basis of more elaborate modelling schemes described later, and utilised throughout the thesis. Finally, some treatment of classical dimensionality reduction techniques is given, as well as a simplified overview of genetic algorithms, which is utilized throughout our work for classifier fusion purposes.

Chapter 3 is a literature review of relevant material to this thesis. It starts by giving an overview of the understanding of human and animal speech perception, and different theories of perception. The chapter then shifts to describing actual systems in literature for category identification from speech, including GID, SID, LID and AID. Particular mention is made of phonotactic systems and acoustic systems. Following this, the chapter goes into some detail on the latest advancements in the field over the last decade, with work on variability compensation, inter-session compensation, joint factor analysis, and the i-Vector paradigm. The chapter ends with an overview of prosody and supra-segmental information extraction from speech.

Chapter 4 is a short description of the different corpora utilised for the experiments in this thesis, namely the 'TIMIT Acoutic-Phonetic Continuous Speech Corpus', the 'ABI-1 Accents of the British Isles Corpus', and the 'WSJCAM0 Cambridge Wall Street Journal Corpus'.

Chapter 5 is a self-contained chapter on our work in GID. We provide an analysis of the main acoustic correlate of gender (pitch). The investigation leads to the idea that there are regions of pitch within each particular gender which together form a more complex model of pitch distribution across both genders. The data suggests that the female population has more of this pitch variability than the male population. In order to apply the right pitch distribution for modelling and classification, we consider the use of MFCCs as providing a codebook for acoustic "context". Within each context, a specific pitch model is built. Pitch from a particular frame can then be classified by first considering the best fitting context from both the male and female trained models, and pitch, as a gender detector, is evaluated only within the pitch distributions for that particular context. Results show how performance is improved, and demonstrate that the system provides robustness to GID for mismatched training and testing corpora. Furthermore, a pitch-shifting mechanism is devised to try and refine GID when the baseline and context-based gender classifiers fail to agree. However, not much improvement over the context-dependent classifier was noticed here.

Our investigation into AID starts in Chapter 6. The first approach assessed is a standard GMM-UBM system for AID, with what is considered to be traditional feature extraction for AID/LID. No form of inter-session compensation is applied to the features. The second approach is to extend the GMM-UBM system with modifications to the feature vectors to include some prosodic information from pitch and first formant. Some performance gains are observed. The third approach extends the standard GMM-UBM system, where each accent class is modelled by multiple GMM-UBM systems, and each of the systems is specific to a particular long-term prosodic context. However, this rather cumbersome extension leads to no gain. In fact, performance deteriorates. The fourth approach makes use of a GMM-SVM system, which is a direct extension of the first GMM-UBM system, with the addition of an SVM classifier to classify supervectors extracted from an adapted UBM per utterance. The accent classes themselves are represented by a single supervector estimated from a GMM trained from all accent training data for the class. The results were surprising in that the classification performance deteriorates considerably. The fifth experiment goes a step back and models each class by multiple supervectors, one per utterance. This technique yields better performance than the GMM-UBM system, but only after dimensionality reduction via PCA and LDA is performed on the supervectors. This is the first indication of how variability due to speaker differences is very problematic in the AID problem.

Chapter 7 is dedicated to our work on AID within the i-Vector framework. We briefly

describe how AID, (similarly to SID and LID) can be approached with this framework. The first approach we present here is to perform basic i-Vector based classification of accents using LDA for dimensionality reduction as well as for classification of the i-Vectors. Performance is reasonably good, and is considerably better than all other previous approaches. However, contrary to what happens in tasks such as SID and LID, the error is still high. In a second approach, we change the classification function to QDA, and observe poorer performance, which is surprising given tat QDA is a more powerful/flexible classifier. This result leads us to try out another approach, this time with support vector machines (which were already shown to perform well on dimensionality reduced supervectors). Though the results obtained are better than for QDA classification, they are not better than those obtained with LDA classification. This suggests an upper limit on classification performance for dimensionality-reduced i-Vectors. At this point, we present a proposal of an iterative-LDA/QDA classifier, where at each iteration of the classification, the weakest accent class is removed, and the LDA projection for a smaller set of classes can be optimized. This classification method leads to some improvements, and is an indication that the separation between classes is still a hard problem for AID. In the second part of the chapter, we assess the positive effect of i-Vector length normalization, which gives some general improvements on classification rates. We then analyze the effect of different dimensionality reduction techniques as alternatives to LDA for speaker variability compensation. Together with multiple i-Vector parameter configurations, we build a fused classifer that gives considerable performance gains, with around 88% AID accuracy for 30 second utterances. Furthermore, we look at how the amount of training data effects classification performance, and observe some gains when each speaker is tested individually, with the rest of the corpus used for training. The optimisations and additions to a standard i-Vector pipeline provide us with a huge leap in performance compared to a standard i-Vector system, and the proposals in this chapter are all intended to mitigate, as far as possible, the effects of speaker variation, which is the main problem for acoustic methods of AID.

In the final part of this thesis (Chapter 8), we have a look at various aspects of AID that are worth investigating now that a very good acoustic AID technique is available. Particularly, we first evaluate what happens when shorter utterances are used. The study revolves around what are standard utterance durations found in literature: 30 seconds, 10 seconds and 3 seconds. Performance degrades on shorter utterances as expected. We then examine the effect of changes to the feature extraction system, where we modify a number of filter-bank parameters. The results show varying degrees of performance, and that the parameters selected for AID so far are not optimal. With these results, we perform an additional layer of fusion, where all

feature extraction combinations with all i-Vector parameter combinations and all projection techniques possible are put together, for a total of 2520 classifiers being considered for fusion. The results are good, especially for short utterances, with 90%, 80% and 57% AID accuracy being reported for 30 second, 10 second and 3 second utterances respectively. The last part of this chapter is about the use of such an AID system for ASR purposes. The work here was done in collaboration with colleagues at the University of Birmingham who were working on adapted models of speech recognition to specific speaker sets and accents. We do not work directly on this aspect in this thesis. However, an early version of our i-Vector AID system with around 80% AID accuracy on 30 second utterances was used as one of the two AID systems tested for model selection in ASR. Our involvement was at the AID level, and in the experimental design, where a number of experiments were designed to set baselines, and a comparison of the effect of different AID systems, together with assumed "correct" accents. The results confirmed that it is indeed very advantageous to perform model selection via AID for ASR systems. The WER gains are impressive. The advantage of using the AID system being proposed is that our system does not rely on any transcription or phone recognition, and is therefore suitable within the ASR task, when the ASR system should decode speech into a sequence of phonemes. Given the improvements in our AID system between the time of these experiments and the completion of this thesis, the usefulness of this scenario is only enhanced further.

9.2 Progress in ABI-1 AID Accuracy

This thesis has focused particularly on applying the i-Vector paradigm to the AID problem. It is good to look at the progress of AID on the ABI-1 corpus over some well known work on the same corpus. This is shown in Table 9.1. The results here contain a different implementation of GMM-UBM and GMM-SVM systems between our work and the work in [4]. Inter-session compensation was not applied to GMM-UBM and GMM-SVM systems in this thesis. Also, our largest GMM-UBM architecture was based on 1024 components and not on 4096 as in [4]. It is very encouraging to see that the best i-Vector system in this work comes very close to the best AID results recorded for the ABI-1 corpus (via ACCDIST), which is based on knowledge of vowel/phoneme transcriptions to work. It is also clear that the i-Vector AID system developed here obtains much better performance than the fused acoustic system in [4], and performs (surprisingly) even better than the acoustic-phonotactic fused system in this same work.

Table 9.1: Comparison of AID results along for the ABI-1 corpus. The most important results are highlighted in **bold**. The asterisk (*) indicates that results are reported only for the SPA passage of each speaker (out of 3 passages in total).

System	AID Accuracy [4] (%) (University of Birmingham)	AID Accuracy (%)
GMM-UBM	56.11	49.47
GMM-SVM	67.72	63.16
GMM-uni-gram	60.12	—
GMM-bi-gram	52.12	—
Acoustic-fused	**73.6**	—
Phonotactics	74.05	—
Acoustic-Phonotactic-fused	**88.8**	—
ACCDIST-Cor.dist.	93.17*	—
ACCDIST-SVM	**95.18***	—
Human	**58.24***	—
i-Vector LDA	—	73.95
i-Vector QDA	—	65.38
i-Vector SVM	—	74.15
i-Vector iter.	—	78.25
i-Vector LDA fused	—	85.96
i-Vector RLDA fused	—	85.50
i-Vector SDA fused	—	85.73
i-Vector NCA fused	—	65.38
i-Vector projection fused	—	87.37
i-Vector frontend/projection fused	—	**90.18**

9.3 Machine Learning the Accents of the British Isles

The previous chapters have discussed in great detail the AID performance and confusion matrices of each test, with final results for different utterance lengths. Another aspect of this work that is worth examining is whether the "machine" has truly learnt something about the accents of the British Isles are, and how they are related to each other. Figure 9.1 gives a visual summary of each accent region, and the top two confusions for the errors in AID accuracy for each accent.

The results can be given some geographical interpretation. The Scottish accents (blue region) report no errors. In the Irish region (green region), ULS is confused with ROI only. ROI in return is also confused with ULS, albeit some confusion with BRM is present. There is only one Welsh accent in the database (red region). The first confusion is with a geographically close BRM accent from the England region (yellow region), and the ULS accent from the Irish region. These two confusions are roughly equidistant in opposite directions. Going into the English region, CRN is confused with SSE/ILO, which are both Southern. Similarly EAN is confused with SSE/CRN. The NCL accent is confused primarily with EYK (the closest region), though

strangely with NWA, which is quite far away. However, the northern nature of the NCL accent may explain the confusion with NWA, and not with any other English accent to the south. The exact same observation can be made for the LAN accent, which was also confused with the EYK and NWA accents. The LVP accent is only confused with the NWA accent, and these are very close to each other. The AID system seems to point out that BRM/ULS are closer to NWA than NWA is to the LVP accent. The EYK accent is confused with LAN (close accent) and BRM. Here again, the results suggest that BRM seems closer to EYK than NCL to EYK.

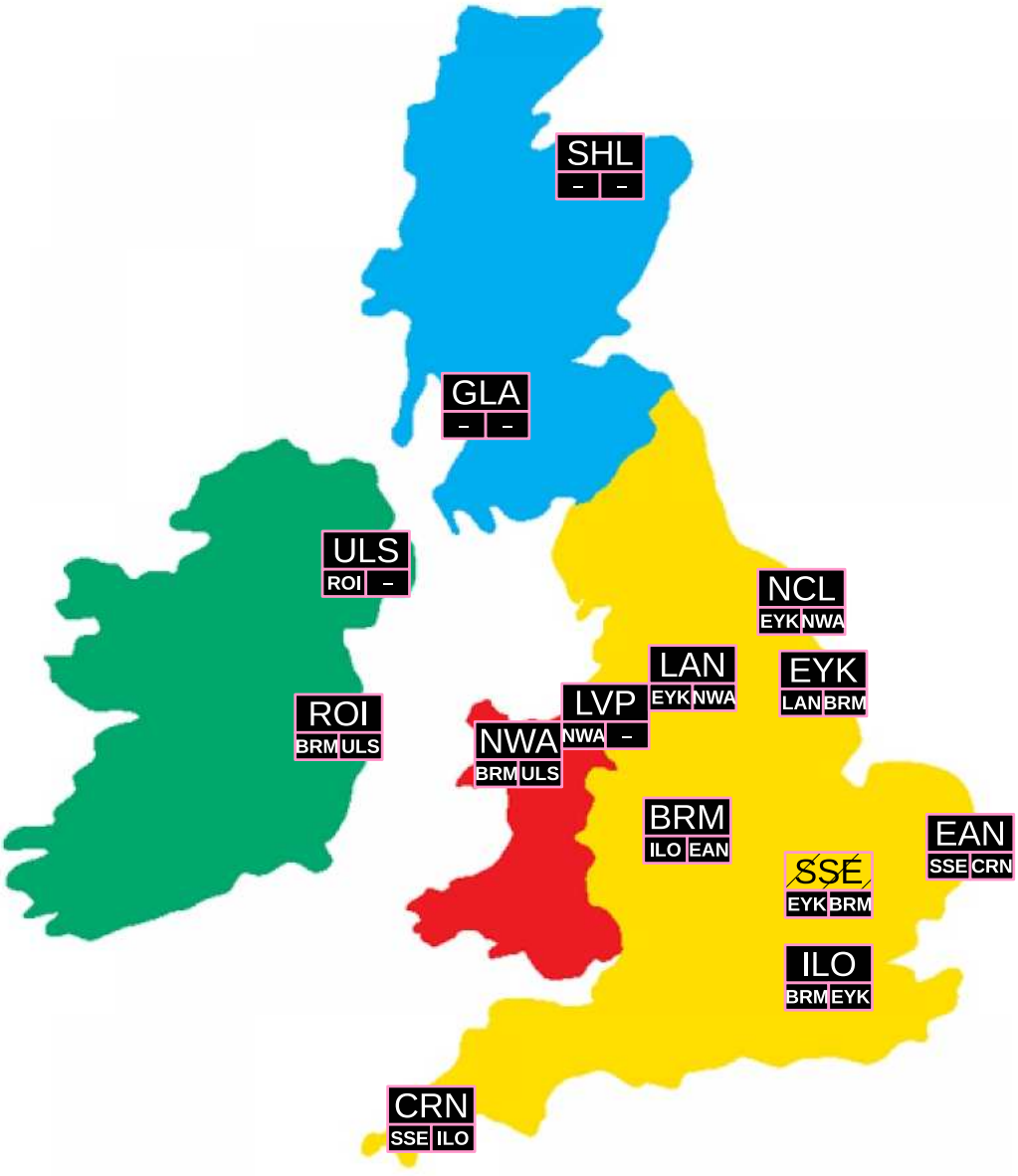


Figure 9.1: The accents of the British Isles as learnt by the AID system in this thesis. Each accent region is marked with the top two accents that have brought about errors in AID classification. Some accents, like SHL, GLA, ULS and LVP have not been confused with any other (100% accuracy), or with only one accent at most. SSE is not tied to a particular region, but as a marker of standard English accent in the south, and is present for reference.

These observations definitely seems to suggest a good correlation of confusions with the actual geographical location and broad regions within the British Isles, and encourage the view that the acoustic AID system has indeed captured, the relative "distance" between accents.

9.4 Future Work

In the introduction to the thesis we listed a number of questions that encompassed the theme and aim of the thesis. We feel we can give a response to each at this point, and discuss how this can lead to future work in the field:

1. We have shown that whilst GID is pretty much considered to be a "solved" problem, there are still ways to enhance performance for mismatched conditions of training and test utterances.

2. We have shown that it is very much possible to achieve very robust AID results without relying on transcriptions and phoneme recognition as part of the frontend to the AID system.

3. The traditional i-Vector system needs to be fine tuned in order to obtain the best performance possible for AID, and the range of performance for various i-Vector systems in this thesis shows how simply applying the traditional i-Vector system to AID is far from sufficient. We have provided an in-depth investigation into this and have provided a guide to building the best i-Vector system for ABI-1 AID.

4. Our investigation has led us to concluded that we can perform very reliable AID using just short-term feature vectors, so long as a number of different frontends, projection, and i-Vector parameters are considered and fused.

There will always be room for improvement for the AID problem. We anticipate that with refinements to the i-Vector model that tackle a number of assumptions about the model, some enhancements will be seen in the future. The main area which would ideally improve is with respect to short duration utterances. Not only do we want AID to perform well on short utterances as a standalone system, but there is great promise of utilizing such a system within the architecture of ASR systems, as we have shown in in the previous chapter.

The ABI-1 corpus was recorded entirely from speakers who had lived all their life in a particular accent region. We expect therefore, that the AID problem will be even harder when

applied to a more realistic scenario in which speakers' accents are made up of various aspects of different accent regions as they migrate from one area to another. In this scenario, AID might usefully be regarded as a continuous-space problem, rather than defined in fixed accent categories as given in the ABI-1 corpus.

Another area that needs investigation, is the relationship between accents and speakers, and how AID and SID technology can be used in tandem to build better and more robust SID systems. We would have liked to investigate this area in this thesis, but there is currently no dataset which we know of that combines a sufficient amount of data of regional accents of a country with enough speakers to pose a problem for current SID algorithms. With regards to the ABI-1 corpus, 100% SID accuracy has already been reported using a baseline GMM-UBM SID system [153] for the standard 30 second utterances. We feel that there is a need to investigate the possibility of whether accents are hard to spoof or not. Spoofing is an area of research in speaker identification and verification. Perhaps the speaking style (partly determined by accent) is hard to spoof in combination with voice mimicry, and robust acoustic AID algorithms could be utilized as part of the solution. This requires the collection of an appropriate corpus which is so far lacking. Provided this is made available at some point, it will give rise to interesting and pertinent avenues of research.

Bibliography

[1] K. Ishizaka and J. Flanagan, "Synthesis of voiced sounds from a two-mass model of the vocal cords," *Bell Systems Technology Journal*, vol. 50, no. 6, pp. 1233–1268, 1972.

[2] J. Joseph P. Campbell, "Speaker recognition: A tutorial," in *Proceedings of the IEEE*, vol. 85, no. 9. IEEE, September 1997, pp. 1441–1443.

[3] M. Najafian, A. DeMarco, S. J. Cox, and M. Russell, "Unsupervised model selection for recognition of regional accented speech," in *Proceedings of the Annual Conference of the International Speech Communication Association*, INTERSPEECH. ISCA, 2014.

[4] A. Hanani, M. J. Russell, and M. J. Carey, "Human and computer recognition of regional accents and ethnic groups from British English speech." *Computer Speech and Language*, vol. 27, no. 1, pp. 59–74, 2013.

[5] J. C. Wells, *Accents of English, Volume 1: The British Isles*. Cambridge University Press, 1982.

[6] T. A. Harley, *The Psychology of Language: From Data To Theory*. Psychology Press, 2008.

[7] J. C. Wells, *Accents of English, Volume 2: The British Isles*. Cambridge University Press, 1982.

[8] A. DeMarco and S. J. Cox, "An accurate and robust gender identification algorithm," in *Proceedings of the Annual Conference of the International Speech Communication Association*, INTERSPEECH. ISCA, 2011, pp. 2429–2432.

[9] ——, "Iterative classification of regional British accents in i-vector space," in *Proceedings of Symposium on Machine Learning in Speech and Language Processing (MLSLP 2012)*, September 2012.

[10] ——, "Native accent classification via i-vectors and speaker compensation fusion," in *Proceedings of the Annual Conference of the International Speech Communication Association*, INTERSPEECH. ISCA, 2013, pp. 1472–1476.

[11] M. Sigmund, *Voice Recognition By Computer*. Tectum Verlag, 2003, ch. 1.

[12] L. Rabiner and B.-H. Juang, *Fundamentals of Speech Recognition*, 1st ed. Prentice-Hall International, April 1993, ch. 2.

[13] G. Fant, *Acoustic Theory of Speech Production*. The Hague: Mouton, 1960.

[14] J. Proakis, *Digital Communications*, 4th ed. McGraw-Hill, August 2000.

[15] J. Harrington and S. Cassidy, *Techniques in Speech Acoustics*, 1st ed. Kluwer Academic Publishers, 1999, ch. 3.

[16] J. Picone, "Signal modeling techniques in speech recognition," in *Proceedings of the IEEE*. IEEE, 1993, pp. 1215–1247.

[17] M. Sigmund, *Voice Recognition By Computer*. Tectum Verlag, 2003, ch. 2.

[18] F. Bimbot, J.-F. Bonastre, C. Fredouille, G. Gravier, I. Magrin-Chagnolleau, SylvainMeignier, T. Merlin, J. Ortega-Garca, D. Petrovska-Delacretaz, and D. A. Reynolds, "A tutorial on text-independent speaker verification," *EURASIP Journal on Applied Signal Processing*, pp. 430–451, 2004.

[19] J. R. Deller, J. H. L. Hansen, and J. G. Proakis, *Discrete-Time Processing of Speech Signals*, 2nd ed. Wiley-IEEE Press, 2000, pp. 225–265.

[20] J. E. Markel and A. H. Gray, *Linear Prediction of Speech*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 1982.

[21] F. Harris, "On the use of windows for harmonic analysis with the discrete fourier transform," in *Proceedings of the IEEE*. IEEE, 1978, pp. 51–84.

[22] J. R. Deller, J. H. L. Hansen, and J. G. Proakis, *Discrete-Time Processing of Speech Signals*, 2nd ed. Wiley-IEEE Press, 2000, pp. 3–98.

[23] L. Rabiner and R. W. Schafer, *Digital Processing of Speech Signals*, 1st ed. Prentice-Hall International, September 1978, ch. 1.

[24] J. Golten, *Understanding Signals and Systems*, 1st ed. McGraw-Hill Publishing Company, 1997, pp. 201–261.

[25] R. W. Ramirez, *The FFT Fundamentals and Concepts*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1985.

[26] X. Huang, A. Acero, and H.-W. Hon, *Spoken Language Processing: a Guide to Theory, Algorithm, and System Development*, 1st ed. Prentice-Hall, 2001, pp. 229–241.

[27] J. O. Smith, *Mathematics of the Discrete Fourier Transform (DFT)*. W3K Publishing, 2007.

[28] J. Golten, *Understanding Signals and Systems*, 1st ed. McGraw-Hill, 1997, ch. 1.

[29] R. I. Damper and J. E. Higgins, "Improving speaker identification in noise by subband processing and decision fusion," *Pattern Recognition Letters*, vol. 24, no. 13, pp. 2167–2173, 2003.

[30] Z. Tufekci and J. Gowdy, "Subband feature extraction using lapped orthogonal transform for speech recognition," in *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, vol. 1, ICASSP. IEEE, May 2001, pp. 149–152.

[31] T. Kinnunen, "Spectral features for automatic text-independent speaker recognition," Licentiate Thesis, Royal Institute of Technology (KTH), Stockholm, Sweden, December 2003.

[32] H. Fastl and E. Zwicker, *Psychoacoustics - Facts and Models*, 3rd ed. Springer, 2007, ch. 3.

[33] S. Molau, M. Pitz, R. Schluter, and H. Ney, "Computing mel-frequency cepstral coefficients on the power spectrum," in *Acoustics, Speech, and Signal Processing Proceedings*, vol. 1, 2001, pp. 73–76.

[34] S. Umesh, L. Cohen, and D. Nelson, "Fitting the mel scale," in *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, ICASSP. Washington, DC, USA: IEEE Computer Society, 1999, pp. 217–220.

[35] J. R. Deller, J. H. L. Hansen, and J. G. Proakis, *Discrete-Time Processing of Speech Signals*, 2nd ed. Wiley-IEEE Press, 2000, pp. 352–408.

[36] D. G. Childers, D. P. Skinner, and R. C. Kemerait, "The cepstrum: A guide to processing," in *Proc. IEEE, Volume 65, p. 1428-1443*, ser. Institute of Electrical and Electronics Engineers, Inc. Conference, vol. 65, 1977, pp. 1428–1443.

[37] X. Huang, A. Acero, and H.-W. Hon, *Spoken Language Processing: a Guide to Theory, Algorithm, and System Development*, 1st ed. Prentice-Hall, 2001, pp. 415–476.

[38] ——, *Spoken Language Processing: a Guide to Theory, Algorithm, and System Development*, 1st ed. Prentice-Hall, 2001, pp. 275–336.

[39] T. Kinnunen and H. Li, "An overview of text-independent speaker recognition: from features to supervectors," *Speech Communication*, vol. 52, no. 1, pp. 12–40, January 2010.

[40] D. A. Reynolds, "Experimental evaluation of features for robust speaker identification," vol. 2, pp. 639–643, October 1994.

[41] L. Rigazio, P. Nguyen, D. Kryze, and J.-C. Junqua, "Separating speaker and environment variabilities for improved recognition in non-stationary conditions," in *7th European Conference on Speech Communication and Technology*. Eurospeech, 2001, pp. 2347–2350.

[42] S. J. Young, D. Kershaw, J. Odell, D. Ollason, V. Valtchev, and P. Woodland, *The HTK Book Version 3.4*. Cambridge University Press, 2006.

[43] M. A. Kohler and M. Kennedy, "Language identification using shifted delta cepstra," in *Circuits and Systems, 2002. MWSCAS-2002. The 2002 45th Midwest Symposium on*, vol. 3, Aug 2002, pp. III–69–72 vol.3.

[44] P. A. Torres-carrasquillo, E. Singer, M. A. Kohler, and J. R. Deller, "Approaches to language identification using Gaussian mixture models and shifted delta cepstral features," in *Proc. ICSLP 2002*, 2002, pp. 89–92.

[45] W. M. Campbell, J. P. Campbell, D. A. Reynolds, E. Singer, and P. A. Torres-Carrasquillo, "Support vector machines for speaker and language recognition," *Computer Speech &, Language*, vol. 20, no. 2-3, pp. 210–229, 2006.

[46] B. S. Atal, "Effectiveness of linear prediction characteristics of the speech wave for automatic speaker identification and verification," *The Journal of the Acoustical Society of America*, vol. 55, no. 6, pp. 1304–1312, 1974.

[47] O. Viikki and K. Laurila, "Cepstral domain segmental feature vector normalization for noise robust speech recognition," *Speech Communication*, vol. 25, no. 1-3, pp. 133–147, 1998.

[48] J. Pelecanos and S. Sridharan, "Feature warping for robust speaker verification," in *2001: A Speaker Odyssey - The Speaker Recognition Workshop*. Crete, Greece: International Speech Communication Association (ISCA), 2001, pp. 213–218.

[49] D. Gerhard, "Pitch extraction and fundamental frequency: History and current techniques," Department of Computer Science, University of Regina, Tech. Rep. 6, November 2003.

[50] U. D. of Phonetics and Linguistics, "Lecture 10: Speech signal analysis," December 2009, http://www.phon.ucl.ac.uk/courses/spsci/matlab/lect10.html.

[51] *A Robust Algorithm for Pitch Tracking (RAPT)*. Speech Coding and Synthesis, 1995, ch. 14.

[52] B. S. Atal, "Automatic speaker recognition based on pitch contours," *The Journal of the Acoustical Society of America*, vol. 52, no. 6B, pp. 1687–1697, 1972.

[53] S. F. (Editor), *Advances in Speech Signal Processing*, 1st ed. Marcel Dekker Inc., 1992, pp. 701–737.

[54] F. Soong, A. Rosenberg, L. Rabiner, and B. Juang, "A vector quantization approach to speaker recognition," vol. 10, no. 2, pp. 387–390, 1985.

[55] T. Kinnunen, T. Kilpelainen, and P. Franti, "Comparison of clustering algorithms in speaker identification," in *Proceedings of the IASTED International Conference of Signal Processing and Communications*. Marbella, Spain: SPC, September 2000, pp. 222–227.

[56] Y. Linde, A. Buzo, and R. Gray, "An algorithm for vector quantizer design," vol. 28, pp. 84–95, January 1980.

[57] N. Nasrabadi and Y. Feng, "Vector quantization of images based upon the Kohonen self-organisation feature maps," vol. 1, pp. 101–108, July 1988.

[58] W. Equitz, "A new vector quantization clustering algorithm," vol. 37, no. 10, pp. 1568–1575, October 1989.

[59] P. Franti, T. Kaukoranta, and O. Nevalainen, "On the splitting method for VQ codebook generation," *Optical Engineering*, vol. 36, no. 11, pp. 3043–3051, November 1997.

[60] P. Franti and J. Kivijarvi, "Randomised local search algorithm for the clustering problem," 2000.

[61] E. Karpov, T. Kinnunen, and n. Pasi Frä, "Symmetric distortion measure for speaker recognition," in *Proceedings of the 9th International Conference on Speech and Computer*. St. Petersburg, Russia: SPECOM, September 2004, pp. 366–370.

[62] D. O'Shaughnessy, *Speech Communications: Human and Machine*, 2nd ed. Wiley-IEEE Press, 1999, pp. 437–458.

[63] D. A. Reynolds, "Automatic speaker recognition using Gaussian mixture models," *Lincoln Laboratory Journal*, vol. 8, no. 2, pp. 173–192, 1995.

[64] D. A. Reynolds and R. Rose, "Robust text-independent speaker identification using Gaussian mixture speaker models," *Speech and Audio Processing, IEEE Transactions on*, vol. 3, no. 1, pp. 72–83, 1995.

[65] D. A. Reynolds, "Speaker identification and verification using Gaussian mixture speaker models," *Speech Communication*, vol. 17, no. 1, pp. 91–108, 1995.

[66] M. Sigmund, *Voice Recognition By Computer*. Tectum Verlag, 2003, ch. 3.

[67] D. A. Reynolds, "A Gaussian mixture modeling approach to text-independent speaker identification," Ph.D. dissertation, Georgia Institute of Technology, Georgia, August 1992, appears in School of Electrical and Computer Engineering Theses and Dissertations Georgia Tech Theses and Dissertations.

[68] K.-H. Yuo and H.-C. Wang, "Joint estimation of feature transformation parameters and Gaussian mixture model for speaker identification," *Speech Commun.*, vol. 28, no. 3, pp. 227–241, 1999.

[69] R. C. Rose, J. Fitzmaurice, E. M. Hofstetter, and D. A. Reynolds, "Robust speaker identification in noisy environments using noise adaptive speaker models," in *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, ICASSP. Washington, DC, USA: IEEE Computer Society, 1991, pp. 401–404.

[70] R. C. Rose and D. A. Reynolds, "Text independent speaker identification using automatic acoustic segmentation," in *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, ICASSP. Albuquerque, NM, USA: IEEE, April 1990.

[71] T. Kinnunen, J. Saastamoinen, V. Hautamaki, M. Vinni, and P. Franti, "Comparative evaluation of maximum a posteriori vector quantization and Gaussian mixture models in speaker verification," *Pattern Recognition Letters*, vol. 30, no. 4, pp. 341 – 347, 2009.

[72] G. Kolano and D. P. Regel-Brietzmann, "Combination of vector quantization and Gaussian mixture models for speaker verification with sparse training data," in *Proceedings of the 6th European Conference on Speech Communication and Technology*. Budapest, Hungary: Eurospeech, September 1999, pp. 1203–1206.

[73] J. Pelecanos, S. Myers, S. Sridharan, and V. Chandran, "Vector quantization based Gaussian modeling for speaker verification," in *ICPR '00: Proceedings of the International Conference on Pattern Recognition*. Washington, DC, USA: IEEE Computer Society, September 2000, pp. 3298–3301.

[74] D. A. Reynolds, T. F. Quatieri, and R. B. Dunn, "Speaker verification using adapted Gaussian mixture models," in *Digital Signal Processing*, January 2000, pp. 10–41.

[75] Yi-Hsiang, W.-H. Tsai, and H.-M. Wang, "Discriminative feedback adaptation for GMM-UBM speaker verification," in *6th International Symposium on Chinese Spoken Language Processing*. Kunming: ISCSLP, December 2008, pp. 1–4.

[76] B. E. Boser, I. M. Guyon, and V. N. Vapnik, "A training algorithm for optimal margin classifiers," in *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*, ser. COLT '92. New York, NY, USA: ACM, 1992, pp. 144–152.

[77] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, no. 3, pp. 273–297, Sep. 1995.

[78] B. Schölkopf, C. Burges, and V. Vapnik, "Extracting support data for a given task," in *First International Conference on Knowledge Discovery and Data Mining*. Menlo Park: AAAI Press, 1995.

[79] L. Ferrer, E. Shriberg, S. Kajarekar, and K. Sonmez, "Parameterization of prosodic feature distributions for SVM modeling in speaker recognition," in *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, vol. 4, ICASSP. Honolulu, Hawaii, USA: IEEE, April 2007, pp. 233–236.

[80] J. Mercer, "Functions of positive and negative type, and their connection with the theory of integral equations," *Philosophical Transactions of the Royal Society, London*, vol. 209, pp. 415–446, 1909.

[81] S. S. Keerthi and C.-J. Lin, "Asymptotic behaviors of support vector machines with gaussian kernel," *Neural Comput.*, vol. 15, no. 7, pp. 1667–1689, Jul. 2003.

[82] H. tien Lin and C.-J. Lin, "A study on sigmoid kernels for svm and the training of non-psd kernels by smo-type methods," Tech. Rep., 2003.

[83] K. Pearson, "On lines and planes of closest fit to systems of points in space," *Philosophical Magazine*, vol. 2, no. 6, pp. 559–572, 1901.

[84] R. A. Fisher, "The use of multiple measurements in taxonomic problems," *Annals of Eugenics*, vol. 7, no. 7, pp. 179–188, 1936.

[85] J. H. Holland, *Adaptation in Natural and Artificial Systems*. Ann Arbor, MI: University of Michigan Press, 1975.

[86] L. Davis, Ed., *Handbook of Genetic Algorithms*. Van Nostrand Reinhold, 1991.

[87] J. E. Baker, "Reducing bias and inefficiency in the selection algorithm," in *Proceedings of the Second International Conference on Genetic Algorithms on Genetic Algorithms and Their Application*. Hillsdale, NJ, USA: L. Erlbaum Associates Inc., 1987, pp. 14–21.

[88] D. Medin, B. H. Ross, and A. B. Markman, *Cognitive Psychology*, 1st ed. Wiley, April 2004, ch. 9.

[89] S. Anderson, E. Shirey, and S. Sosnovsky, "Speech perception," Department of Information Science, University of Pittsburgh, Tech. Rep., 2003, http://www.sis.pitt.edu/~anderson/downloads/sma-speech-perception.pdf.

[90] W. J. Hardcastle and N. Hewitt, *Coarticulation: Theory, Data and Techniques*, new ed. Cambridge University: Cambridge University Press, November 2006, ch. 1, cambridge Studies in Speech Science & Communication.

[91] P. Lieberman and S. E. Blumstein, *Speech Physiology, Speech Perception, and Acoustic Phonetics*, 1st ed. Cambridge University: Cambridge University Press, February 1988, ch. 7.

[92] D. G. MacKay, G. Wulf, C. Yin, and L. Abrams, "Relations between word perception and production - new theory and data on the verbal transformation effect," *Journal of Memory and Language*, vol. 32, no. 5, pp. 624–646, 1993.

[93] R. B. Ivry and T. C. Justus, "A neural instantiation of the motor theory of speech perception," *Trends in Neurosciences*, vol. 24, no. 9, pp. 513–515, September 2001.

[94] R. R. Benson, B. S. D. H. Whalen, Matthew Richardson, V. P. Clark, S. Lai, and A. M. Liberman, "Parametrically dissociating speech and nonspeech perception in the brain using fMRI," *Brain and Language*, vol. 78, pp. 364–396, 2001.

[95] D. Medin, B. H. Ross, and A. B. Markman, *Cognitive Psychology*, 1st ed. Wiley, April 2004, ch. 7.

[96] M. D. Hauser, C. Miller, D. Morris, and J. Mehler, "Language discrimination by human newborns and by cotton-top tamarin monkeys," *Science*, vol. 288, no. 5464, pp. 349–351, April 2000.

[97] P. Lobacz, *Processing and decoding the signal in speech perception*. John Benjamins Pub Co, January 1986.

[98] A. Francis and H. Nusbaum, "Paying attention to speaking rate," in *Proceedings of the Fourth International Conference on Spoken Language*, ICSLP. The University of Chicago: Center for Computational Psychology, October 1996, pp. 1537–1540.

[99] H. McGurk and J. MacDonald, "Hearing lips and seeing voices," *Nature*, vol. 264, pp. 746–748, December 1976.

[100] K. P. Green, "Studies of the McGurk effect: Implications for theories of speech perception," in *Proceedings of the Fourth International Conference on Spoken Language*. University of Arizona: ICSLP, October 1996, pp. 1652–1655.

[101] L. Bernstein and C. Benoit, "For speech perception by humans or machines, three senses are better than one," in *Proceedings of the Fourth International Conference on Spoken Language*. House Ear Institute, California and Universit Stendhal, Grenoble: ICSLP, October 1996, pp. 1477–1480.

[102] K. Johnson, E. Strand, and M. D'Imperio, "Auditory-visual integration of talker gender in vowel perception," *Journal of Phonetics*, vol. 27, no. 4, pp. 359–384, October 1999.

[103] M. Zissman, "Comparison of four approaches to automatic language identification of telephone speech," *Speech and Audio Processing, IEEE Transactions on*, vol. 4, no. 1, pp. 31–, Jan 1996.

[104] P. Matejka, P. Schwarz, J. Cernocký, and P. Chytil, "Phonotactic language identification using high quality phoneme recognition." in *Proceedings of the Annual Conference of the International Speech Communication Association*, INTERSPEECH. ISCA, 2005, pp. 2237–2240.

[105] H. Li and B. Ma, "A phonotactic language model for spoken language identification," in *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*. Ann Arbor, Michigan: Association for Computational Linguistics, June 2005, pp. 515–522.

[106] L. Wang, E. Ambikairajah, and E. H. C. Choi, "Robust language identification based on fused phonotactic information with MLKSFM pre-classifier," in *Multimedia and Expo, 2009. ICME 2009. IEEE International Conference on*, June 2009, pp. 121–124.

[107] M. Heck, S. Stuker, and A. Waibel, "A hybrid phonotactic language identification system with an SVM back-end for simultaneous lecture translation," in *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, ICASSP. IEEE, March 2012, pp. 4857–4860.

[108] M. F. BenZeghiba, J.-L. Gauvain, and L. Lamel, "Phonotactic language recognition using MLP features." in *Proceedings of the Annual Conference of the International Speech Communication Association*, INTERSPEECH. ISCA, 2012.

[109] W. Liu, W. Zhang, Z. Li, and J. Liu, "Parallel absolute-relative feature based phonotactic language recognition." in *Proceedings of the Annual Conference of the International Speech Communication Association*, INTERSPEECH. ISCA, 2013, pp. 59–63.

[110] H.-S. Lee, Y.-C. Shih, H.-M. Wang, and S.-K. Jeng, "Subspace-based phonotactic language recognition using multivariate dynamic linear models." in *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, ICASSP. IEEE, 2013, pp. 6870–6874.

[111] H. Suo, M. Li, P. Lu, and Y. Yan, "Using SVM as back-end classifier for language identification," *EURASIP J. Audio Speech Music Process.*, vol. 2008, pp. 2:1–2:6, Jan. 2008.

[112] W. M. Campbell, J. P. Campbell, D. A. Reynolds, D. A. Jones, and T. R. Leek, "Phonetic speaker recognition with support vector machines." in *NIPS*, S. Thrun, L. K. Saul, and B. Schölkopf, Eds. MIT Press, 2003.

[113] A. Hanani, M. J. Carey, and M. J. Russell, "Improved language recognition using mixture components statistics." in *Proceedings of the Annual Conference of the International Speech Communication Association*, T. Kobayashi, K. Hirose, and S. Nakamura, Eds., INTERSPEECH. ISCA, 2010, pp. 741–744.

[114] G. Zipf, "Human behaviour and the principle of least-effort." Cambridge, MA: Addison-Wesley, 1949.

[115] J.-L. Gauvain, A. Messaoudi, and H. Schwenk, "Language recognition using phone latices." in *Proceedings of the Annual Conference of the International Speech Communication Association*, INTERSPEECH. ISCA, 2004.

[116] E. Singer, P. A. Torres-Carrasquillo, T. P. Gleason, W. M. Campbell, and D. A. Reynolds, "Acoustic, phonetic, and discriminative approaches to automatic language identification." in *Proceedings of the Annual Conference of the International Speech Communication Association*, INTERSPEECH. ISCA, 2003.

[117] V. Hubeika, L. Burget, P. Matejka, and P. Schwarz, "Discriminative training and channel compensation for acoustic language recognition," in *Proceedings of the Annual Conference of the International Speech Communication Association*, INTERSPEECH. ISCA, 2008, pp. 301–304.

[118] L. Burget, P. Matejka, and J. Cernocky, "Discriminative training techniques for acoustic language identification," in *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, ICASSP. IEEE, 2006, pp. 209–212.

[119] M. Zissman, "Automatic language identification using Gaussian mixture and hidden Markov models," in *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, vol. 2, ICASSP. IEEE, April 1993, pp. 399–402 vol.2.

[120] W. Campbell, T. Gleason, J. Navratil, D. Reynolds, W. Shen, E. Singer, and P. Torres-Carrasquillo, "Advanced language recognition using cepstra and phonotactics: MITLL system performance on the NIST 2005 language recognition evaluation," in *Speaker and Language Recognition Workshop, 2006. IEEE Odyssey 2006: The*, June 2006, pp. 1–8.

[121] W. Zhang, B. Li, D. Qu, and B. Wang, "Automatic language identification using support vector machines," in *Signal Processing, 2006 8th International Conference on*, vol. 1, 2006, pp. 16–20.

[122] F. Castaldo, D. Colibro, E. Dalmasso, P. Laface, and C. Vair, "Acoustic language identification using fast discriminative training." in *Proceedings of the Annual Conference of the International Speech Communication Association*, INTERSPEECH. ISCA, 2007, pp. 346–349.

[123] S. Kullback and R. A. Leibler, "On information and sufficiency," *Ann. Math. Statist.*, vol. 22, no. 1, pp. 79–86, 1951.

[124] N. Dehak, "Discriminative and generative approaches for long- and short-term speaker characteristics modeling: Application to speaker verification," Ph.D. dissertation, 2009, aAINR50490.

[125] M. N. Do, "Fast approximation of Kullback-Leibler distance for dependence trees and hidden Markov models," *Signal Processing Letters, IEEE*, vol. 10, no. 4, pp. 115–118, Mar. 2003.

[126] T. Jaakkola and D. Haussler, "Exploiting generative models in discriminative classifiers." in *Advances in Neural Information Processing Systems 11 – NIPS Conference, Denver, Colorado, USA, November 30 - December 5, 1998)*, M. J. Kearns, S. A. Solla, and D. A. Cohn, Eds. The MIT Press, 1999, pp. 487–493.

[127] S. Fine, J. Navratil, and R. Gopinath, "A hybrid GMM/SVM approach to speaker identification," in *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, vol. 1, ICASSP. IEEE, 2001, pp. 417–420.

[128] V. Wan and S. Renals, "SVMSVM: support vector machine speaker verification methodology," in *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, vol. 2, ICASSP. IEEE, April 2003, pp. II–221–4.

[129] W. Campbell, "Generalized linear discriminant sequence kernels for speaker recognition," in *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, vol. 1, ICASSP. IEEE, May 2002, pp. I–161–I–164.

[130] W. Campbell, D. Sturim, D. Reynolds, and A. Solomonoff, "SVM based speaker verification using a GMM supervector kernel and nap variability compensation," in *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, vol. 1, ICASSP. IEEE, May 2006, pp. I–I.

[131] W. Campbell, D. Sturim, and D. Reynolds, "Support vector machines using GMM supervectors for speaker verification," *Signal Processing Letters, IEEE*, vol. 13, no. 5, pp. 308–311, May 2006.

[132] N. Dehak and G. Chollet, "Support vector GMMs for speaker verification," in *Speaker and Language Recognition Workshop, 2006. IEEE Odyssey 2006: The*, June 2006, pp. 1–4.

[133] K. Wu and D. G. Childers, "Gender recognition from speech. part i: Coarse analysis," *The Journal of the Acoustical Society of America*, vol. 90, no. 4, pp. 1828–1840, 1991.

[134] J. Fussell, "Automatic sex identification from short segments of speech," in *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, vol. 1, ICASSP. IEEE, Apr 1991, pp. 409–412.

[135] E. Parris and M. Carey, "Language independent gender identification," *Acoustics, Speech, and Signal Processing, IEEE International Conference on*, vol. 2, pp. 685–688, 1996.

[136] Y.-M. Zeng, Z.-Y. Wu, T. Falk, and W.-Y. Chan, "Robust GMM based gender classification using pitch and RASTA-PLP parameters of speech," in *Machine Learning and Cybernetics, 2006 International Conference on*, 2006, pp. 3376 –3379.

[137] S. Slomka and S. Sridharan, "Automatic gender identification optimised for language independence," in *TENCON '97. IEEE Region 10 Annual Conference. Speech and Image Technologies for Computing and Telecommunications., Proceedings of IEEE*, vol. 1, Dec 1997, pp. 145–148 vol.1.

[138] D. Tran and D. Sharma, "Automatic gender recognition," in *Proceedings of the 2nd WSEAS International Conference on Electronics, Control and Signal Processing*. Stevens Point, Wisconsin, USA: World Scientific and Engineering Academy and Society (WSEAS), 2003, pp. 49:1–49:5.

[139] M. Pronobis and M. Magimai.-Doss, "Analysis of f0 and cepstral features for robust automatic gender recognition," Idiap, Idiap-RR Idiap-RR-30-2009, November 2009.

[140] P. Kumar, N. Jakhanwal, A. Bhowmick, and M. Chandra, "Gender classification using pitch and formants," in *Proceedings of the 2011 International Conference on Communication, Computing & Security*, ser. ICCCS '11. New York, NY, USA: ACM, 2011, pp. 319–324.

[141] R. Vergin, A. Farhat, and D. O'Shaughnessy, "Robust gender-dependent acoustic-phonetic modelling in continuous speech recognition based on a new automatic male/female classification," in *In Fourth Internalional Conference on Spoken Language Processing*, 1996, pp. 1081–1084.

[142] H. Harb and L. Chen, "Gender identification using a general audio classifier," in *Proceedings of the 2003 International Conference on Multimedia and Expo - Volume 1*, ser. ICME '03. Washington, DC, USA: IEEE Computer Society, 2003, pp. 733–736.

[143] M. Zissman, T. Gleason, D. Rekart, and B. Losiewicz, "Automatic dialect identification of extemporaneous conversational, Latin American Spanish speech," in *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, vol. 2, ICASSP. IEEE, May 1996, pp. 777–780.

[144] D. Miller and J. Trischitta, "Statistical dialect classification based on mean phonetic features," in *Spoken Language, 1996. ICSLP 96. Proceedings., Fourth International Conference on*, vol. 4, Oct 1996, pp. 2025–2027 vol.4.

[145] "Language accent classification in American English," *Speech Communication*, vol. 18, no. 4, pp. 353 – 367, 1996.

[146] M. Lincoln, S. Cox, and S. Ringland, "A comparison of two unsupervised approaches to accent identification." in *ICSLP*. ISCA, 1998.

[147] P. Angkititrakul and J. Hansen, "Advances in phone-based modeling for automatic accent classification," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 14, no. 2, pp. 634–646, March 2006.

[148] R. Huang, J. Hansen, and P. Angkititrakul, "Dialect/accent classification using unrestricted audio," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 15, no. 2, pp. 453–464, Feb 2007.

[149] M. Huckvale, "Accdist: An accent similarity metric for accent recognition and diagnosis." in *Speaker Classification (2)*, ser. Lecture Notes in Computer Science, C. MÃijller, Ed., vol. 4441. Springer, 2007, pp. 258–275.

[150] F. S. Richardson, W. M. Campbell, and P. A. Torres-Carrasquillo, "Discriminative n-gram selection for dialect recognition." in *Proceedings of the Annual Conference of the International Speech Communication Association*, INTERSPEECH. ISCA, 2009, pp. 192–195.

[151] F. Biadsy, J. Hirschberg, and N. Habash, "Spoken Arabic dialect identification using phonotactic modeling," in *In Proceedings of EACL 2009 Workshop on Computational Approaches to Semitic Languages*, 2009.

[152] F. Biadsy, J. Hirschberg, and M. Collins, "Dialect recognition using a phone-GMM-supervector-based SVM kernel." in *Proceedings of the Annual Conference of the International Speech Communication Association*, T. Kobayashi, K. Hirose, and S. Nakamura, Eds., INTERSPEECH. ISCA, 2010, pp. 753–756.

[153] S. Safavi, A. Hanani, M. Russell, P. Jancovic, and M. Carey, "Contrasting the effects of different frequency bands on speaker and accent identification," *Signal Processing Letters, IEEE*, vol. 19, no. 12, pp. 829–832, Dec 2012.

[154] P. Zhan and A. Waibel, "Vocal tract length normalization for large vocabulary continuous speech recognition," Carnegie Mellon University, Tech. Rep., 1997.

[155] D. Klatt, "A digital filter bank for spectral matching," in *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, vol. 1, ICASSP. IEEE, Apr 1976, pp. 573–576.

[156] M. J. F. Gales and S. Young, "An improved approach to the hidden Markov model decomposition of speech and noise," in *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, vol. 1, ICASSP. IEEE, Mar 1992, pp. 233–236.

[157] R. Auckenthaler, M. Carey, and H. Lloyd-Thomas, "Score normalization for text-independent speaker verification systems," *Digital Signal Processing*, vol. 10, no. 1-3, pp. 42–54, 2000.

[158] D. Sturim and D. Reynolds, "Speaker adaptive cohort selection for Tnorm in text-independent speaker verification," in *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, vol. 1, ICASSP. IEEE, March 2005, pp. 741–744.

[159] D. Ramos-Castro, J. Fiérrez-Aguilar, J. Gonzalez-Rodriguez, and J. Ortega-Garcia, "Speaker verification using speaker- and test-dependent fast score normalization." *Pattern Recognition Letters*, vol. 28, no. 1, pp. 90–98, 2007.

[160] R. Vogt, B. Baker, and S. Sridharan, "Modelling session variability in text-independent speaker verification." in *Proceedings of the Annual Conference of the International Speech Communication Association*, INTERSPEECH. ISCA, 2005, pp. 3117–3120.

[161] L. Heck and M. Weintraub, "Handset-dependent background models for robust text-independent speaker recognition," in *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, vol. 2, ICASSP. IEEE, Apr 1997, pp. 1071–1074.

[162] R. Dunn, T. Quatieri, D. Reynolds, and J. Campbell, "Speaker recognition from coded speech and the effects of score normalization," in *Signals, Systems and Computers, 2001. Conference Record of the Thirty-Fifth Asilomar Conference on*, vol. 2, Nov 2001, pp. 1562–1567 vol.2.

[163] L. Burget, P. Matejka, P. Schwarz, O. Glembek, and J. Cernocky, "Analysis of feature extraction and channel compensation in a GMM speaker recognition system," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 15, no. 7, pp. 1979–1986, Sept 2007.

[164] P. Kenny, P. Ouellet, N. Dehak, V. Gupta, and P. Dumouchel, "A study of interspeaker variability in speaker verification," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 16, no. 5, pp. 980–988, July 2008.

[165] D. Reynolds, "Channel robust speaker verification via feature mapping," in *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, vol. 2, ICASSP. IEEE, April 2003, pp. II–53–6.

[166] C. Vair, D. Colibro, F. Castaldo, E. Dalmasso, and P. Laface, "Channel factors compensation in model and feature domain for speaker recognition," in *Speaker and Language Recognition Workshop, 2006. IEEE Odyssey 2006: The*, June 2006, pp. 1–6.

[167] A. Hanani, "Human and computer recognition of regional accents and ethnic groups from British English speech," Ph.D. dissertation, 2012.

[168] P. Kenny and P. Dumouchel, "Disentangling speaker and channel effects in speaker verification," in *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, vol. 1, ICASSP. IEEE, May 2004, pp. I–37–40.

[169] N. Brummer, "Spescom datavoice NIST 2004 system description," *Proceedings of the NIST Speaker Recognition Evaluation*, 2004.

[170] P. Kenny, "Joint factor analysis of speaker and session variability: Theory and algorithms," Tech. Rep., 2005.

[171] C. M. Bishop, *Pattern Recognition and Machine Learning (Information Science and Statistics)*, 1st ed. Springer, 2007.

[172] P. Kenny, G. Boulianne, P. Ouellet, and P. Dumouchel, "Joint factor analysis versus eigenchannels in speaker recognition," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 15, no. 4, pp. 1435–1447, May 2007.

[173] ——, "Speaker and session variability in GMM-based speaker verification," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 15, no. 4, pp. 1448–1460, May 2007.

[174] P. Kenny, G. Boulianne, and P. Dumouchel, "Eigenvoice modeling with sparse training data," *Speech and Audio Processing, IEEE Transactions on*, vol. 13, no. 3, pp. 345–354, May 2005.

[175] H. Lei, "Joint factor analysis (JFA) and i-vector tutorial," International Computer Science Insitute, Tech. Rep., October 2011.

[176] N. Dehak, R. Dehak, P. Kenny, N. Brummer, P. Ouellet, and P. Dumouchel, "Support vector machines versus fast scoring in the low-dimensional total variability space for speaker verification." in *Proceedings of the Annual Conference of the International Speech Communication Association*, INTERSPEECH. ISCA, 2009, pp. 1559–1562.

[177] N. Dehak, P. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front-end factor analysis for speaker verification," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 19, no. 4, pp. 788–798, May 2011.

[178] A. O. Hatch, S. S. Kajarekar, and A. Stolcke, "Within-class covariance normalization for SVM-based speaker recognition." in *Proceedings of the Annual Conference of the International Speech Communication Association*, INTERSPEECH. ISCA, 2006.

[179] W. Campbell, D. Sturim, D. Reynolds, and A. Solomonoff, "SVM based speaker verification using a GMM supervector kernel and nap variability compensation," in *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, vol. 1, ICASSP. IEEE, May 2006, pp. I–I.

[180] D. M. González, O. Plchot, L. Burget, O. Glembek, and P. Matejka, "Language recognition in ivectors space." in *Proceedings of the Annual Conference of the International Speech Communication Association*, INTERSPEECH. ISCA, 2011, pp. 861–864.

[181] N. Dehak, P. A. Torres-Carrasquillo, D. A. Reynolds, and R. Dehak, "Language recognition via i-vectors and dimensionality reduction." in *Proceedings of the Annual Conference of the International Speech Communication Association*, INTERSPEECH. ISCA, 2011, pp. 857–860.

[182] L. F. D'Haro, O. Glembek, O. Plchot, P. Matejka, M. Soufifar, R. de Córdoba, and J. Cernocký, "Phonotactic language recognition using i-vectors and phoneme posteriogram counts." in *Proceedings of the Annual Conference of the International Speech Communication Association*, INTERSPEECH. ISCA, 2012.

[183] A. Kanagasundaram, D. Dean, J. Gonzalez-Dominguez, S. Sridharan, D. Ramos, and J. Gonzalez-Rodriguez, "Improving short utterance based i-vector speaker recognition using source and utterance-duration normalization techniques." in *Proceedings of the Annual Conference of the International Speech Communication Association*, F. Bimbot, C. Cerisara, C. Fougeron, G. Gravier, L. Lamel, F. Pellegrino, and P. Perrier, Eds., INTERSPEECH. ISCA, 2013, pp. 2465–2469.

[184] A. Ikeno and J. H. L. Hansen, "The effect of listener accent background on accent perception and comprehension," *EURASIP J. Audio, Speech and Music Processing*, vol. 2007, 2007.

[185] F. Ramus and J. Mehler, "Language identification with suprasegmental cues: A study based on speech resynthesis," *Journal of the Acoustical Society of America*, vol. 105, no. 1, pp. 512–521, 1999.

[186] D. E. Callan, J. A. Jones, A. M. Callan, and R. Akahane-Yamada, "Phonetic perceptual identification by native- and second-language speakers differentially activates brain regions involved with acoustic phonetic processing and those involved with articulatory-auditory/orosensory internal models," *NeuroImage*, vol. 22, no. 3, pp. 1182 – 1194, 2004.

[187] E. McDermott and A. Nakamura, "Production-oriented models for speech recognition." *IEICE Transactions*, vol. 89-D, no. 3, pp. 1006–1014, 2006.

[188] T. Toda, A. W. Black, and K. Tokuda, "Acoustic-to-articulatory inversion mapping with Gaussian mixture model." in *Proceedings of the Annual Conference of the International Speech Communication Association*, INTERSPEECH. ISCA, 2004.

[189] P. K. Ghosh and S. Narayanan, "A generalized smoothness criterion for acoustic-to-articulatory inversion," *Journal of the Acoustical Society of America*, vol. 128, no. 4, pp. 2162–2172, October 2010.

[190] P. K. Ghosh and S. S. Narayanan, "A subject-independent acoustic-to-articulatory inversion." in *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, ICASSP. IEEE, 2011, pp. 4624–4627.

[191] F. Pellegrino, J.-H. Chauchat, and R. Rakotomalala, "Can automatically extracted rhythmic units discriminate among languages?" in *Speech Prosody*, 2002, pp. 562–565.

[192] J.-L. Rouas, J. Farinas, F. Pellegrino, and R. André-Obrecht, "Rhythmic unit extraction and modelling for automatic language identification." *Speech Communication*, vol. 47, no. 4, pp. 436–456, 2005.

[193] P. F. Dominey and F. Ramus, "Neural network processing of natural language: I. Sensitivity to serial, temporal and abstract structure of language in the infant," *Language and Cognitive Processes*, vol. 15, no. 1, pp. 87–127, 2000.

[194] J. Farinas and R. André-Obrecht, "Identification automatique des langues : variations sur les multigrammes," in *XXIIIéme Journées d'Etude sur la Parole (JEP'2000) , Aussois, France*. ICP-GFCP-SFA-ISCA, Juin 2000, pp. 373–376.

[195] F. Ramus, M. Nespor, and J. Mehler, "Correlates of linguistic rhythm in the speech signal," *Cognition*, vol. 75, no. 1, pp. AD3 – AD30, 2000.

[196] M. Barkat, J. Ohala, and F. Pellegrino, "Prosody as a distinctive feature for the discrimination of Arabic dialects." in *Eurospeech*. ISCA, 1999.

[197] M. Beckman, M. Diaz-Campos, J. T. McGory, and T. Morgan, "Intonation across Spanish, in the tones and break indices framework." *International Journal of Latin and Romance Linguistics*, vol. 14, no. 1, pp. 9–36, January 2006.

[198] J. S. Garofolo, L. F. Lamel, W. M. Fisher, J. G. Fiscus, D. S. Pallett, and N. L. Dahlgren, "DARPA TIMIT acoustic phonetic continuous speech corpus CDROM," 1993.

[199] S. D'Arcy, J. Russell, S. Browning, and M. Tomlinson, "The Accents of the British Isles (ABI) Corpus," in *Modelisations pour l'Identification des Langues. MIDL Paris*, 2005, pp. 115–119.

[200] T. Robinson, J. Fransen, D. Pye, J. Foote, and S. Renals, "WSJCAM0: A British English, speech corpus for large vocabulary continuous speech recognition," vol. 1, Detroit, 1995, pp. 81–84.

[201] A. Stolcke, S. S. Kajarekar, L. Ferrer, and E. Shrinberg, "Speaker recognition with session variability normalization based on MLLR adaptation transforms." *IEEE Transactions on Audio, Speech and Language Processing*, vol. 15, no. 7, pp. 1987–1998, 2007.

[202] M. Ferras, C.-C. Leung, C. Barras, and J.-L. Gauvain, "Comparison of speaker adaptation methods as feature extraction for SVM-based speaker recognition." *IEEE Transactions on Audio, Speech and Language Processing*, vol. 18, no. 6, pp. 1366–1378, 2010.

[203] M. Brookes, "VOICEBOX: Speech Processing tool for MATLAB," 2014. [Online]. Available: http://www.ee.ic.ac.uk/hp/staff/dmb/voicebox/voicebox.html

[204] W. Groen, L. van Orsouw, M. Zwiers, S. Swinkels, R. van der Gaag, and J. Buitelaar, "Gender in voice perception in autism," *Journal of Autism and Developmental Disorders*, vol. 38, pp. 1819–1826, 2008.

[205] O. Parviainen, "Sound Touch Audio Processing Library."

[206] J. Sohn, N. S. Kim, and W. Sung, "A statistical model-based voice activity detection," *Signal Processing Letters, IEEE*, vol. 6, no. 1, pp. 1–3, Jan 1999.

[207] L. Mary and B. Yegnanarayana, "Extraction and representation of prosodic features for language and speaker recognition," *Speech Communication*, vol. 50, pp. 782–796, October 2008.

[208] N. Dehak, P. Dumouchel, and P. Kenny, "Modeling prosodic features with joint factor analysis for speaker verification," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 15, no. 7, pp. 2095–2103, Sept 2007.

[209] M. Kockmann, L. Ferrer, L. Burget, E. Shriberg, and J. Cernocky, "Recent progress in prosodic speaker verification," in *Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on*, May 2011, pp. 4556–4559.

[210] D. Garcia-Romero and C. Y. Espy-Wilson, "Analysis of i-vector length normalization in speaker recognition systems," in *Proceedings of the Annual Conference of the International Speech Communication Association*, INTERSPEECH. ISCA, 2011, pp. 249–252.

[211] G. C. Cawley and N. L. Talbot, "On over-fitting in model selection and subsequent selection bias in performance evaluation," *J. Mach. Learn. Res.*, vol. 11, pp. 2079–2107, Aug. 2010.

[212] J. H. Friedman, "Regularized Discriminant Analysis," *Journal of the American Statistical Association*, vol. 84, no. 405, pp. 165–175, 1989.

[213] D. Cai, X. He, and J. Han, "Semi-supervised discriminant analysis," in *Proceedings of the International Conference on Computer Vision (ICCV'07)*, 2007.

[214] J. Goldberger, S. Roweis, G. Hinton, and R. Salakhutdinov, "Neighbourhood components analysis," in *Advances in Neural Information Processing Systems 17*. MIT Press, 2004, pp. 513–520.

[215] W. Rao and M.-W. Mak, "Boosting the performance of i-vector based speaker verification via utterance partitioning." *IEEE Transactions on Audio, Speech & Language Processing*, vol. 21, no. 5, pp. 1012–1022, 2013.

[216] P. Trudgill, *The Dialects of England*. Blackwell Publishers Inc., 1999.

[217] S. Goronzy, *Robust Adaptation to Non-Native Accents in Automatic Speech Recognition*, ser. Lecture Notes in Computer Science. Springer, 2002, vol. 2560.

[218] J. J. Humphries and P. C. Woodland, "Using accent-specific pronunciation modelling for improved large vocabulary continuous speech recognition." in *Eurospeech*, G. Kokkinakis, N. Fakotakis, and E. Dermatas, Eds. ISCA, 1997.

[219] M. Tjalve and M. Huckvale, "Pronunciation variation modelling using accent features." in *Proceedings of the Annual Conference of the International Speech Communication Association*, INTERSPEECH. ISCA, 2005, pp. 1341–1344.

[220] H. Kamper, F. J. M. Mukanya, and T. Niesler, "Multi-accent acoustic modelling of South African English," *Speech Communication*, vol. 54, no. 6, pp. 801 – 813, 2012.

[221] Y. Deng, X. Li, C. Kwan, B. Raj, and R. Stern, "Continuous feature adaptation for non-native speech recognition," *International Journal of Computer, Information Science and Engineering*, vol. 1, no. 6, pp. 164 – 171, 2007.

[222] Y. Zheng, R. Sproat, L. Gu, I. Shafran, H. Zhou, Y. Su, D. Jurafsky, R. Starr, and S.-Y. Yoon, "Accent detection and speech recognition for Shanghai-accented Mandarin." in *Proceedings of the Annual Conference of the International Speech Communication Association*, INTERSPEECH. ISCA, 2005, pp. 217–220.

[223] M. Bacchiani, "Rapid adaptation for mobile speech applications." in *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, ICASSP. IEEE, 2013, pp. 7903–7907.

[224] O. Siohan and M. Bacchiani, "Ivector-based acoustic data selection." in *Proceedings of the Annual Conference of the International Speech Communication Association*, F. Bimbot, C. Cerisara, C. Fougeron, G. Gravier, L. Lamel, F. Pellegrino, and P. Perrier, Eds., INTERSPEECH. ISCA, 2013, pp. 657–661.