SPEAKER IDENTIFICATION AND VERIFICATION USING

GAUSSIAN MIXTURE MODELS

ATHANASIOS VALSAMAKIS

Department of Computer Science

APPROVED:

Yannis Stylianou Associate Professor

Panagiotis Tsakalidis Associate Professor

Giorgos Tziritas Professor

Dimitris Plexousakis

Chairman of the Graduate Studies Committee

to my

MOTHER and FATHER

SPEAKER IDENTIFICATION AND VERIFICATION USING GAUSSIAN MIXTURE MODELS

by

ATHANASIOS VALSAMAKIS

THESIS

Presented to the Faculty of the Computer Science Department of

The University of Crete at Iraklio

in Partial Fulfillment of

the Requirements for

the Degree of

MASTER OF SCIENCE

Department of Computer Science

THE UNIVERSITY OF CRETE

April 2006

Acknowledgements

I would like to express my deep-felt gratitude to my adviser, Dr. Yannis Stylianou of the Computer Science Department at The University of Crete in Iraklio, for his advice, encouragement and enduring patience.

I also wish to thank the other members of my laboratory, Giorgos Kafentzis, Yannis Pantazis, Miltos Vasilakis, Ioannis Agiomirianakis, Maria Markaki, Andre Holzapfel, Yannis Sfakianakis, Giorgos Tsagarakis, Panagiotis Tripolitis and Xristos Tsagarakis. Their suggestions, comments and constant help were invaluable to the completion of this work. As a special note, I would like to thank Sascha Schimke for the helpful collaboration we have while I visited University of Magdeburg, Germany, last summer.

Additionally, I want to thank Computer Science Department professors and staff for all their hard work and dedication, providing me the means to complete my degree and prepare for a career as a computer scientist.

And finally, I must thank my dear fiancée, Sofia for putting up with me during the development of this work with continuing, loving support.

Abstract

Automatic speaker verification and identification are probably the most natural and economical methods for solving the problems of unauthorized use of computer and communications systems and multilevel access control. With the ubiquitous telephone network and microphones embedded into computers, the cost of a speaker recognition system might only be for the software of the recognition algorithm. Biometric systems automatically recognize a person using distinguishing traits. Speaker recognition is a performance biometric i.e. you perform a task to be recognized. Your voice, like other biometrics, cannot be forgotten or misplaced, unlike knowledge-based (e.g. password) or possession-based (e.g. key) access control methods. Due to the inherent variability of the speech signal, as far as the identity of a person is concerned, we emphasize the use of statistical approaches in this thesis for speaker recognition. Speaker recognition is based on appropriate probabilistic models. The form of the Gaussian mixture model (GMM) motivates its use as a representation of speakers for text-independent speaker recognition. The most popular method for automatic speaker recognition uses the cepstrum, with nonlinear frequency axis following the Bark or mel scale. Using these features the performance of the system is quite satisfactory. One step forward is the use of sub-cepstrum which gives even better results with the same computational cost.

Table of Contents

Page

Acknowledgements iv

Abstract v

Table of Contents vi

List of Tables ix

List of Figures x

1 Introduction 1

1.1 Principles of Speaker Recognition 1

1.2 Open Questions about Speaker Recognition 4

1.3 Scope of this Thesis 9

2 Background 11

2.1 Motivation 11

2.2 Introduction 11

2.3 Verification versus Identification 14

2.4 Recognition Techniques 15

2.4.1 Model Evaluation 16

2.4.2 Text Dependent versus Text Independent 17

2.4.3 Statistical versus Dynamic Features 18

2.4.4 Vector Quantization 19

2.4.5 Stochastic Models 19
2.4.6 Neural Network Approaches 20
2.4.7 Similarity and Distance Measures 20
2.5 Features that Distinguish Speakers 22
2.5.1 Spectral Features 23
2.5.2 Cepstral Analysis 24
2.5.3 High-level Features 25
2.6 System Design 25
2.6.1 Data Collection 26
2.6.2 Multi-stage Recognition 27
2.6.3 Effects of Different Communication Channels 28
2.7 Speaker Recognition by Humans 29
3 Speaker Recognition System Design 31
3.1 Introduction 31
3.2 Bayes' Theorem in General 32
3.3 Probability Density 33
3.3.1 Parametric Techniques 34
3.3.2 Maximum Likelihood 35
3.4 Mixture Models 37
3.5 Estimation via Maximum Likelihood 40
3.5.1 Multivariate Gaussian Mixture Density Estimation 44
3.6 Speech Analysis 45
3.6.1 Short-time Speech Analysis 46

3.6.2 Windowing 47

3.6.3 Frequency-domain Parameters 49

3.6.4 Filterbank Analysis 50

3.6.5 Short-time Fourier Transform Analysis 50

3.7 Cepstral Coefficients 52

3.7.1 Mel-frequency Ceptrum 54

3.7.2 Sub-Cepstrum 56

4 Experimental Results and Implementation Details 59

4.1 Introduction 59

4.2 Speech Database 60

4.3 Speaker Dependent Features 60

4.3.1 Extraction of Cepstral Coefficients 60

4.3.2 Extraction of Sub-cepstral Coefficients 62

4.4 Speaker Models 65

4.5 Identification System 65

4.6 Verification System 67

4.7 Identification Under Adverse Conditions 68

5 Concluding Remarks 70

5.1 Conclusion 70

5.2 Future Work 71

References 72

List of Tables

3.1 The Bark frequency scale. 53

4.1 Results for the identification system, the test speech is about 2 *sec*. 66

4.2 Results for the identification system under adverse conditions. 69

List of Figures

3.1 Surface plot representing the Gaussian probability density for the case of two dimensional feature vectors. 36

3.2 Curve representing a probability density function composed of a mixture of Gaussian functions. 41

3.3 Log-likelihood behavior for four different speaker models in the speaker recognition system. 43

3.4 (a) Hamming window of 20 *ms* duration sampled by a sampling frequency of 16 *kHz.* (b) A speech frame of 20 *ms* for analysis. 49

3.5 Mel filter bank frequency responses. (a) Normalized $\bar{H}_m[k]$ and (b) Unnormalized $H_m[k]$. 55

3.6 Cepstral coefficients extracted from two different speech frames. 57

3.7 Sub-cepstral coefficients extracted from two different speech frames. 58

4.1 Impulse responses for the 2nd, 3rd, 10th and 13th filter in the filterbank. . 63

4.2 Energies from mel-scale and subband filter banks, (a) mel-scale filter energy from filter number 14 ($\approx$ 3100 Hz), (b) subband filter energy from the same filter number. 64

4.3 (a) Verification using mel-cepstrum, (b) Verification using sub-cepstrum. . 68

Chapter 1

Introduction

1.1 Principles of Speaker Recognition

In this section we introduce recent advances and assess the current status in speaker recognition technology. Although many recent advances and successes have been achieved in speaker recognition, there are still many problems for which good solutions remain to be found.

While speech recognition is the process of extracting the linguistic message underlying a spoken utterance, speaker recognition is the process of automatically recognizing who is speaking by using speaker-specific information inherent in the speech wave [8]. Using this technology we can identify someone requesting access to a computer system. Namely, speaker recognition is concerned with identifying the person speaking an utterance. As speech interaction with computers becomes more pervasive, the utility of automatically identifying a person increases. Applications that someone can bring in mind are wide ranging, including voice dialing, telephone financial transactions (e.g. banking over the telephone, telephone shopping), database access, voice mail, security control for confidential information, and remote access to computers. Another important application of speaker recognition technology is the use for forensic purposes.

The general area of speaker recognition can be divided into two categories, *speaker identification* and *speaker verification*. In speaker identification the goal is to determine which of a group of known voices best matches the input voice sample. That is, we pick out in some way a speaker, from a set of registered speakers, that is *closest* to the input speaker voice. In the different task of speaker verification, the goal is to decide whether a speaker corresponds to a particular known voice or to some other unknown voice. In other words, accepting or rejecting the identity claim of a speaker. A speaker unknown to the system who is posing as a known speaker is called an impostor. For example, a person attempts to log on a computer by entering a user name and then is prompted to say a phrase. The input voice sample is compared to a reference of the claimed speaker's voice and the system either accepts the unknown speaker and allows further access or rejects the unknown speaker as an impostor and denies access.

The fundamental difference between identification and verification is the number of decision alternatives. In identification, the number of decision alternatives is equal to the size of the population (i.e. the number of registered speakers) whereas in verification, there are only two choices, accept or reject, regardless of the population size. That is, in verification we compare the input voice sample with the voice within our database and if this goes beyond a specified threshold we accept the speaker otherwise we reject them. As a consequence, speaker identification performance decreases as the population size increases, whereas speaker verification performance theoretically approaches a constant, independent of the size of the population, unless the distribution of physical characteristics of each speaker is extremely biased. In general, there are two types of error in a speaker verification system, namely false acceptance, where an impostor is accepted as a known speaker, and

false rejection where a known speaker is rejected as an impostor. The former error is more undesirable than the latter one, because in a false rejection, we can kindly ask for new try.

The speaker recognition task can also be divided into *text-dependent* and *text-independent*. In a text-dependent system the speech used to train and test the system is forced to be the same word or phrase. In text-independent system the training and testing speech are different and can be any word or phrase the user likes. The main objective of this thesis is to develop a text-independent speaker recognition system capable of high recognition accuracy for short utterances. The text-dependent methods are usually based on template-matching techniques where the input speech sample and the reference sample or reference model of the registered speaker are aligned, and the similarity between the two is extracted [5, 11]. Based on the similarity between the two, we make a decisison. In general, text-dependent recognition methods achieve higher recognition scores than text-independent methods. This is the fact, since in text-independent methods there is greater variability between input and reference speech.

However, there are several applications, such as surveillance or forensic applications, in which predetermined keywords cannot be used. Furthermore, human beings have the ability to recognize a speaker irrespective of what he is saying. So, text-independent methods are more preferable in order to reveal the mechanism of human hearing. In consequence, text-independent methods have recently attracted more attention in the scientific community.

Both text-independent and text-dependent speaker recognition methods suffer from being vulnerable to fraudulent activities. That is, these systems can easily fail, because someone who plays back the recorded voice of a registered speaker saying a keyword or a sentence to a microphone can be accepted as the registered speaker. To cure this problem,

there are many suggestions. One solution is the following: each user is prompted to utter a randomly chosen phrase every time they want to gain access to the system [13]. Yet, this method is not reliable enough, since it can fail with advanced electronic recording equipment that can reproduce phrases in a requested manner. Another more secure and reliable solution could use, in parallel, other biometric data, such as the face or the signature of a person, so that to make a decision using a combination of all these information. Therefore, fusion methods of different biometric data have recently been proposed for person identification [14–16].

1.2 Open Questions about Speaker Recognition

Although many recent advances and successes have been achieved, there are still many problems for which good solutions remain to be found. Some of the major problems are discussed in what follows:

- *How can human beings correctly recognize speakers ?*

Research on the mechanism of how human beings receive and analyze an acoustic signal has been conducted for many years, but little knowledge about the hearing capability of humans being has been extracted. Which exactly acoustic parameters do listeners utilize for making their judgments is still a mystery for scientific community. Despite that, some progress have been made in this direction. One of the results found by experiments is that segmental (spectral envelope) information plays a more important role than supra-segmental (pitch and energy) information.

- *Is it useful to study the mechanism of speaker recognition by human beings ?*

Do human auditory models help improve speaker recognition performance? An example of how knowledge of human hearing characteristics have led to a new idea for automatic speaker recognition was the creation of cepstral parameters. These parameters is a representation defined as the real cepstrum of a windowed short-time signal derived from the Fourier transform of the signal. These parameters were proposed based on experimental results. The important point is that a nonlinear frequency scale can be used, which approximates the behavior of the human auditory system. Although the number of such examples is very small, we expect further advancements in our understanding of the mechanism of speech perception by humans, to help create new engineering ideas in the future.

- *Is it useful to study physiological mechanism of speech production to get new ideas for speaker recognition ?*

Hitherto, we don't understand how the identity of a person is encoded in their voice. Furthermore, how this knowledge can help us towards the automatic speaker recognition. Another interesting question is how perceptually identical voices can be produced by our vocal organs.

- *What feature parameters are appropriate for speaker recognition ?*

In selecting acoustic attributes for automatic speaker recognition, we want the features to reflect the unique characteristics of a speaker. Mel-cepstral features [17] exploit the auditory principles as well as the decorrelating properties of the cepstrum. As such, the

mel-cepstrum has proven to be one of the most successful feature representation in speech-related tasks. These features are exactly the same as those used for speech recognition systems. In spite of their success, it is worth of searching for a new set of feature parameters that is more appropriate for speaker recognition than the present set. Here, we have a paradox, because the same set of features have been successfully used for speech recognition tasks and speaker recognition tasks, even though in the first situation we need speaker-independent features and in the second situation we need speaker-dependent features. Yet, in both cases they work satisfactorily.

- *How can we fully exploit the clearly evident encoding of identity in prosody and other high-level features of speech ?*

There are a variety of voice attributes that characterize a speaker. We categorize speaker-dependent voice characteristics as high-level and low-level. High-level attributes include "clarity", "roughness", "magnitude", and "animation". Other high-level attributes are "prosody", i.e. pitch intonation and articulation rate, and "dialect". This kind of features can be used as perceptual cues in determining the identity of a speaker. Despite their usefulness, these attributes can be difficult to extract by machine for automatic speaker recognition. On the other hand, low-level attributes, such as cepstral coefficients, are more measurable. The question is how can we measure such high-level features with the use of a computer.

- *Is it possible to make clear which people in a particular group account for the majority of errors in a speaker recognition system ?*

Is there any universal set of parameters, algorithms etc. that performs good for any set of speakers? Or the encoding of identity is so speaker-dependent that the optimal parameters, algorithms, acoustic features depend on the speaker set to be identified? Will speaker recognition be practically 100% of the speaker population or will some speakers need to be excluded because their exceptional identity encoding scheme cannot be exploited with the particular system implementation? Is there any set of parameters that is good for separating speakers whose voices sound identical, such as twins? Can we separate this identically sounding voices? Is it worth trying or we don't have any chance because even people face difficulties in this situation?

- *Can we ever reliably cluster speakers on the basis of similarity or dissimilarity ?*

Is there any kind of similarity or dissimilarity relationship between speakers and if yes, how can we measure it? Can we make absolute statements that these speakers will never be confused by an automatic speaker recognition system?

- *How do we acquire realistically sized databases for evaluation purposes ?*

How should we choose the speaker set within the database? How should we select the texts i.e. sentences, words, syllables? How should we make common speech databases for evaluating speaker recognition techniques? It is crucial to have common speech databases for comparing the effectiveness of different techniques. A major speech database designed for speaker recognition and related tasks is the *Switchboard* corpus [18]. It is also important to consider the recording conditions, such as the background noise, the type of microphone etc. in order to collect a database suitable for calculating the quality, robustness and value of the various techniques.

- *How do we deal with long-term variability in people's voices ?*

How often should we update the speaker models to cope with gradual changes in people's voice? How we adequately retain the speaker models for long-term variability?

- *Can we model or develop strategies for dealing with factors that significantly alter a person's voice ?*

How do we cope with situations of illness of a person (such as flu, sore throat etc.) that significantly alters their voice? How do we deal with circumstances of emotion and fatigue. Until this time, no one has succeeded in modeling these voice alterations.

- *How can we deal with deliberately disguised voices in fraudulent claims?*

Are there acoustic features that are invariant regardless of the disguise method? This question is of particular importance in forensic speaker recognition. Criminals may disguise their voices or mimic another person's voice. Does this pose a real threat to our speaker recognition systems? How can we deal successfully with this problem?

- *Is speech better than other biometric data for person identification purposes ?*

Fingerprints, signatures, iris, face and various other means can also be used for person identification. In contrast to fingerprinting, we probably have to assume that there are pairs of speakers whose voices cannot be separated when the population is large. This is true, but someone would argue against this statement that the fingerprint can be easily copied with some kind of apparatus. This is a trade-off between physiological and behavioral traits of human beings that we will discuss later in this writing.

- *What is the specified standard of a speaker recognition system in order for them to be utilized in the real world ?*

Up to now, experiments have shown that speaker recognition systems have to be used in combination with other biometrics (such as face, signature etc.) in order for them to reach the standards that a real world application dictates.

- *Is it a good idea to combine speech and speaker recognition in order to improve the performance of the system ?*

A text-prompted method is an example of a system that utilizes speech recognition in order to improve the accuracy of a speaker recognition system. Since semantic information and speaker dependent information are highly correlated in the speech signal, combining the techniques and ideas of speech and speaker recognition is expected to improve recognition performance.

There have been many satisfactory answers to mostly all of the aforementioned questions. However, there are still many problems that good solutions remain to be found in the field. It is also important to develop methods to cope with the problems of distortion due to background and channel noises as we will see in the following chapters.

1.3 Scope of this Thesis

The main ambition of this thesis is to design and implement a robust text-independent speaker identification and verification system. The problems of distortion of the speaker-dependent characteristics due to interfering background noise and the loss of accuracy when

the population size increases are thoroughly contemplated. In order to achieve this objective, we have used an effective statistical speaker representation which accomplishes outstanding speaker identification and verification results. For this purpose, we have used *the Gaussian mixture speaker models* for text-independent speaker identification that demonstrate high identification accuracy.

In selecting the acoustic features of each speaker, we have used firstly the mel-cepstral coefficients that has proven to be one of the most successful feature representation in speech-related recognition tasks. But because the limited temporal resolution of the aforementioned features, we have alternatively used the sub-cepstal coefficients that shows better recognition accuracy.

This thesis is organized as follows. Chapter 2 provides a background examination of the automatic speaker identification. Chapter 3 introduces some mathematical principles that the speaker recognition system is based on. Chapter 4 presents experimental results and some details about the implementation. Finally, Chapter 5 gives some concluding remarks and future research directions.

Chapter 2

Background

2.1 Motivation

Automatic speaker verification and identification are probably the most natural and economical methods for solving the problems of unauthorized use of computer and communications systems and multilevel access control. With the ubiquitous telephone network and microphones embedded into computers, the cost of a speaker recognition system might only be for the software of the recognition algorithm. Biometric systems automatically recognize a person using distinguishing traits. Speaker recognition is a performance biometric i.e. you perform a task to be recognized. Your voice, like other biometrics, cannot be forgotten or misplaced, unlike knowledge-based (e.g. password) or possession-based (e.g. key) access control methods. Speaker recognition systems can be made somewhat robust against noise and channel variations, ordinary human voice changes (e.g. minor colds), and mimicry by humans and tape recorders.

2.2 Introduction

Computer analysis of speech has many purposes. Speech coding and speech recognition are the major applications of speech analysis. In this thesis we deal with a fundamental

application of speech analysis: automatic speaker or voice recognition. The purpose of this chapter is to provide a background review of the area of automatic speaker recognition and to introduce notation and nomenclature used throughout the thesis.

In speech recognition, variation due to different speakers in the speech signal corresponding to the same spoken text was viewed as some kind of "noise" to be either eliminated or not giving attention for the analysis procedure. When the task is to identify *who* is talking rather than *what* is said, speech must be processed to extract attributes of speaker variability instead of semantic attributes [3, 4, 7]. Compared to an automatic speech recognition system, there has been less research for speaker recognition because fewer applications exist and it is less understood about which speech aspects characterize a speaker than about semantic information inherent in the conversational speech signal. There are two main speaker recognition applications: firstly, verifying a person's identity prior to admission to a secure computer system or to a transaction over the telephone and secondly, for forensic purposes in police work. While fingerprints or retinal scans are usually more reliable ways to verify a claimant, voice identification has the convenience of easy data collection over the telephone and voice is more difficult to be copied because it is a stochastic (random) process. Many companies providing limited access to computer databases would like to include automatic customer services over the telephone. Since personal number codes typed on the keypad or physical keys can be lost, stolen, or forgotten, speaker recognition provides a practical alternative.

In automatic speech recognition, much is known about the speech production process linking a text and a corresponding speech signal. The corresponding acoustic events have been well studied. For speaker recognition, however, the acoustic aspects of what character-

izes the differences between voices are unclear and difficult to separate from signal aspects that convey semantic information. There are three sources of variation among speakers: physical differences in vocal folds and vocal tract shape, differences in speaking style, such as speaking rate, and differences in in what speakers choose to say. Automatic speaker recognizers exploit only the first two variation sources examining low-level acoustic features of speech [2, 8] since a speaker's tendency to use certain word and syntactic structures is difficult to measure and easy to mimic.

Without prevention by that, there are no acoustic cues specifically or exclusively related to speaker's identity. Most of the parameters and features in speech analysis, as we will see later in this text, contain information useful for the identification of both the speaker and the spoken message. The two types of information, however, are coded in the speech signal quite differently. Unlike automatic speech recognition, where decisions are made for every phoneme, speaker identification usually requires only one decision based on the entire utterance. It is very difficult to find acoustic attributes that reliably distinguish speakers. Usually, speakers recognition techniques utilize statistics averaged over the entire utterance. Statistical averaged methods are often used in text-independent speaker recognition where training and testing employ different phrases. While in text-dependent applications the same utterance is used for training and testing. Although, only one decision is made in speaker recognition, the set of choices can vary depending on the application. Most practical applications need only a binary decision, i.e. is the speaker who he claims to be?, but some other ask which of N stored voices best matches a test input voice.

2.3 Verification versus Identification

There are two related but different types of voice recognition: *automatic speaker verification* [1,19] or authentication and *automatic speaker identification* [4,20]. Both use a stored database of reference models for N known speakers, and similar analysis and decision techniques are employed. Automatic speaker verification is the simpler task since it requires only comparing the test pattern with one reference model and a binary decision whether the test speech matches the model of the claimant, although this depends on the method used. In verification on one hand, speakers known to the system are called "customers", while unregistered speakers are called impostors as we saw in Section 1.1. In identification, on the other hand, requires choosing which of N known voices best matches an input voice. Since N comparisons and decisions are often necessary, the error rate increases with N for identification, while verification may have error rates independent of N [2], depending on the method used.

We distinguish closed-set and open-set speaker recognition. The former means that only enrolled speakers are considered. The latter means that the voice of the test speaker may not be among the N registered speaker, in which case a "no match" decision should be made. Since most real life applications require reliability, another option for both identification and verification would be to delay the decision in the case of uncertainty and ask the speaker to provide more speech. While speaker verification is much more common than identification, identification meets many useful applications. For example, segmenting conversational speech into speaker turns [21].

While the worst case performance for both verification and identification systems is 0%

correct, if we think a bit more, we will find that it is 50% for verification, since we have a binary decision, but only $\frac{1}{N} \cdot 100\%$ for identification, assuming that each speaker is equally likely. There are two categories of errors: *false acceptance* and *false rejection*. In the former, the system incorrectly accepts an impostor during verification or identifies a wrong person during identification. In the latter, the system rejects a true claimant in verification or incorrectly finds "no match" in identification. The decision to accept or reject usually depends on a threshold, that is, if the similarity between a test and a reference model exceeds the threshold (i.e. a model likelihood is too low), the system rejects the request.

Depending on the cost of each type of error, systems can be designed to minimize an overall cost by biasing the decision in favor of less costly outcomes. Low thresholds are usually preferred because false acceptance is often less desirable i.e. admitting an impostor to a secure system might be extremely bad, while rejecting a registered speaker is usually only annoying. Many researchers adjust system parameters so that the two types of error occur equally often.

2.4 Recognition Techniques

Analysis techniques are quite similar for speech and speaker recognition. Data reduction through modeling (parameterization) and feature extraction are important for speaker identification/verification for reasons of efficiency. Template matching, distance measures, and stochastic models have been some common techniques for speaker identification/verification. The models used for speaker recognition emphasize speaker characteristics rather than semantic or other information. The memory used grows linearly with the

number of speakers (the population size). Although, feature-based speaker recognition has small memory requirements but is usually more difficult to implement.

2.4.1 Model Evaluation

Since typical speech parameters and features contain information about phonemes and the speaker, some speaker recognition systems use methods identical to those for speech recognition, with the difference that the models are created for speakers rather than for phonemes or words. In general such systems store information for every speaker. Therefore, for a speaker identification or verification system it is necessary to build a model of the voice of each speaker and then to store the parameters of this model that represents each registered speaker. For this purpose, we need to use speaker-dependent features extracted from the speech waveform. For example, the oral and nasal tract length during different sounds, the vocal fold mass and shape, and the location and size of the false vocal folds, if accurately measured from the speech waveform, could be used as features in a physiological speaker model. We call this the training phase of the recognition system, and the corresponding speech data used in building a speaker model is called the training data. During the recognition or testing phase, we then "match" the features measured from the waveform of a test speech sample (test data) against reference speaker models obtained during training. In practice, however, it is difficult to derive speaker anatomy from only the speech signal. Rather, it is typical to use acoustic spectral features derived from the speech waveform, as we will see later in this text.

2.4.2 Text Dependent versus Text Independent

Evaluating test utterances for speaker identity is much simpler when the test and training utterance is constrained to be the same word or phrase. But this is only applicable when we have cooperative speakers, who train the system and later test it with the same words. This text-dependent method, using the same text for training and testing, is rarely used for speaker identification or verification. For example, in forensic work, speakers are usually uncooperative, test and training speech are not the same. Having different text for training and testing, the text-independent method can still use some kind of model matching, but the information that must be stored in for the models is much different. Error rates for text-independent recognition are higher than comparable text-dependent cases [22]. To achieve good performance for text-independent recognition, much more speech data are usually needed for both training and testing than for text-dependent recognition. Training often uses 30 seconds per speaker and test utterances are usually no more than 5 seconds. On the other hand, the performance of text-dependent systems is highly correlated with the vocabulary that is chosen. Furthermore, text-dependent speaker recognition systems may have more security problems. For example, stolen recordings of a registered speaker could be used to deceive the text-dependent system, whereas a text-independent system could request different phrases for each recognition test. Recordings could still be used to deceive the system but in this case, they would be less successful.

2.4.3 Statistical versus Dynamic Features

Since acoustic cues of a speaker's identity exist in the speech signal, some systems utilize methods of averaged parameters rather than exploit the full-time sequence. This global approach is most useful in text-dependent speaker recognition. The simplest approach conceptually takes long-term averages of speech parameters over all available data from each speaker to yield one mean vector, which is used as reference. Long-term parameters can produce good recognition accuracy for normal speech and even for speech spoken under stress, but not for disguised speech. But the long test utterances needed to obtain long-term averages usually preclude real-time applications. Furthermore, long-term features are often sensitive to variations due to transmission channels (e.g. telephone). They do not also take account of other important speaker-specific information in the speech signal.

Since recognition using long-term statistics is often impractical for real-time text independent applications, another approach in such cases involves identifying specific sounds in the test utterance and comparing them with stored sounds for each speaker. Specific phonemes extracted from each speaker's training data, segmented manually, if necessary, are stored for reference. At recognition time, each utterance of unknown text and speaker is scanned to locate phonemes corresponding to those in memory. Distance measures between patterns of each test phoneme and each corresponding reference are averaged over all the located phonemes in the test utterance to yield an overall outcome. This outcome is then compared against a threshold in verification applications or the minimum outcome over all speakers is picked in speaker identification cases.

2.4.4 Vector Quantization

Since automatic segmentation of continuous speech into phonemes is difficult, speaker recognition based on analysis specific phonemes is not common. Alternative techniques for speaker recognition attempt to compare corresponding test and reference phonemes without explicitly locating them in a test utterance. Vector quantization (VQ) is one such technique applied to both speech and speaker recognition. The data-reduction property of VQ has been very important for speech coding application and has also proved very useful for speaker recognition. Besides avoiding segmentation of the speech signal and allowing short test utterances, VQ is computationally efficient compared with storing and comparing large amounts of template data. Thus VQ paved the way for successful speaker identification or verification. Speaker recognition via VQ can yield satisfactory results with relatively short test utterances. But a comparison of VQ and HMM methods for speaker recognition showed that HMM's were superior [23]. Within HMM's, the state transitions provided little information for text-independent applications.

2.4.5 Stochastic Models

Stochastic models have largely replaced the use of templates and long-term averages over the past few years for speaker verification/identification. The Hidden Markov Model (HMM) method have been successfully applied for speaker recognition. It has been developed lately a set of HMM's for each speaker. With enough training 1% error rate is possible. The aforementioned approach have been emerged from speech recognition techniques.

Recently, single-state stochastic models have become popular for speaker verification

or identification. *Gaussian Mixture Models* GMM's are quite similar to HMM-based approaches, but omit the time information inherent in HMM's. Since speaker identification or verification makes only a single decision, no distinction is needed among the frames of each speaker utterance. Typically, 10 Gaussians using full covariance matrices provides good performance. For the training and testing phase, 30 seconds and 5 seconds of speech, respectively, have given very good results [4, 14, 16].

2.4.6 Neural Network Approaches

Neural network methods have been used with limited success for speaker identification and verification [30]. Instead of simply training neural networks for each speaker, the neural networks are trained to model differences among known speakers, which allows a smaller number of parameters. Neural network methods are similar in complexity and performance with VQ recognizers. The main disadvantage is the need to retrain the entire network for each new speaker.

2.4.7 Similarity and Distance Measures

To match templates or to calculate VQ distortions, there are different kinds of distance measures that have been used for speaker recognition. Some of the most common distances are the Euclidean and Mahalanobis that have often yielded good results. Other measures such as correlations and city-block distances are sometimes used but have given inferior results. Recall that the Mahalanobis distance between two M-dimensional vectors x and y

is

$$d(x, y) = \sqrt{(x - y)^T W^{-1} (x - y)} \tag{2.1}$$

The W matrix allows different weighting for the M vector parameters. This distance has come from statistical decision theory. For speaker recognition, each utterance may be viewed as a point in the M-dimensional space and the utterances for each speaker describe a multivariate probability density function in that space. That is each speaker is a random generator of points in an M-dimensional space. Assuming speaker identification among equally likely speakers (i.e. equal prior probabilities), Bayes's rule specifies choosing the speaker whose density is most likely to have generated the test utterance. Points of equal Mahalanobis distance from y form a hyper ellipsoid centered at y, whose principal axes and lengths are the eigenvectors and eigenvalues of W^{-1}, respectively [24].

Because the difficulty of estimating density functions from a limited amount of training data, most speaker recognizers assume a parametric form of density such as a Gaussian, which can simply and fully described by a mean vector μ and a covariance matrix W. In this simplified situation, that is when we assume that the speech parameters used for speaker identification have unimodal distributions resembling Gaussians, the density of a feature vector x for speaker i would be

$$f_i(x) = \frac{1}{(2\pi)^{-\frac{M}{2}} |W_i|^{-\frac{1}{2}}} \exp\left\{ -\frac{1}{2}(x - \mu_i) W_i^{-1} (x - \mu_i) \right\} \tag{2.2}$$

where $|W_i|$ is the determinant of the covariance matrix W_i. In simplified system, it can be used a fixed W matrix for all speakers, instead of individual W_i. This could be done because, firstly it is difficult to obtain accurate estimates for W_i matrix especially when there are limited training data of individual speakers, secondly, using one W saves memory

and thirdly, W_i may be similar for different speakers. Despite that, recognition accuracy may increase in some systems by using individual W_i when examining each speaker's density as we will see later in this text.

Given a test feature vector x for speaker identification, speaker j is selected as the correct speaker if

$$f_j(x) > f_i(x) \quad \text{for all speakers} \;\; i \neq j \tag{2.3}$$

After some calculations Equation (2.3) reduces to minimizing the Mahalanobis distance of Equation (2.1), using μ_i in place of y.

Another distance measure used for speaker identification is the Bhattacharyya distance [25]. For two speakers i and j, whose feature distributions have mean vectors μ_i and μ_j, and covariance matrices W_i and W_j, the Bhattacharyya distance is given by the following formula

$$d^2 = \frac{1}{2}\log\left(\frac{|W_i + W_j|}{\sqrt{|W_i||W_j|}}\right) + \frac{1}{8}(\mu_i - \mu_j)^T \left(\frac{W_i + W_j}{2}\right)^{-1} (\mu_i - \mu_j) \tag{2.4}$$

The first term depends only on the W matrices, while the second resembles the Mahalanobis distance.

2.5 Features that Distinguish Speakers

For simplicity, most speaker recognition systems use standard speech parameters such as LPC coefficients or bandpass filter-bank energies. Viewing speaker identification as a problem of separating probability densities in an M-dimensional space, we can obtain better results and lower computational cost by more careful selection of the parameters or features

that make up the space. Ideally, the space should use a few independent features that have similar small intra-speaker variances and large inter-speaker variances, which lead to compact widely separated clusters for individual speakers [31]. In practical terms, the features should also be easy to measure, be stable over time, change little in different environments, and not be susceptible to mimicry.

The two sources of speaker diversity, physiological and behavioral, lead to two types of useful features. Physiological features are relatively fixed for a speaker and depend on the anatomy of the speaker's vocal tract. While they can be affected by health conditions (e.g. colds that block the nasal cavity), physiological features are less susceptible to mimicry of impostors than behavioral features. While behavioral features can be used to distinguish people with similar vocal tracts, impostors usually find it easier to deceive speaker recognizers that are based on behavioral features than those using physiological features. Mimics usually imitate the global speaking rate well, but not fine spectral details.

2.5.1 Spectral Features

Spectral features tend to be very useful for speaker identification and verification. Vowels, nasals and fricatives are often recommended for speaker identification because they are relatively easy to identify in the speech signal and their spectra contain features that reliably distinguish speakers. Nasals have been of particular interest because the nasal cavities of different speakers are distinctive and are not easily modified, except in cases of colds. The difficulty in locating nasals in the speech signal has hindered their application in speaker identification systems. Furthermore, the difficulty and computational cost of phoneme seg-

mentation have led many recognizers to avoid examining specific sounds and to use general spectral attributes in the recognition process. In addition, one of the major problems with the spectral features is that they are very sensitive to additive noise distortion. Addition of white noise to the speech signal affects the speech power spectrum at all the frequencies, but the effect is less noticeable in the higher amplitude portions of the spectrum.

2.5.2 Cepstral Analysis

LPC parameters (e.g. reflection coefficients) have been used, in the past few years, for speaker identification and verification, but recently, researchers have shown that cepstral coefficients are superior [26]. Besides being invariant to spectral distortion (e.g. telephone channel distortion) cepstral coefficients yield high recognition accuracy. Excellent results have been reported through GMM techniques [4, 10, 16].

The process used for calculating the cepstral coefficients is by segmenting the speech signal into overlapping frames of about 20 *ms* and then computing a vector of cepstral coefficients for each frame, as we will see later in this text. The mean value for each coefficient over each frame is subtracted from each coefficient function, this is called cepstral mean subtraction (CMS). This technique yields a signal that minimizes environmental and intraspeaker variability. Transitional information, in the form of delta coefficients, have been reported that increases recognition accuracy, especially in cases of channel mismatch between testing and training [27]. Another technique known as RASTA processing [28] eliminates very slowly varying attributes of the speech signal. It removes not only convolutional channel distortion, but also some speaker-dependent characteristics, and as a result

does not help with additive channel noise and nonlinearities. Long-term average subtraction is inappropriate because much speaker-dependent information would be removed [29].

2.5.3 High-level Features

In addition to the low-level spectral features used by current systems, there are many other sources of speaker information in the speech signal that can be used. These include word usage, prosodic measures, and other long-term signal measures. This work will be aided by the increasing use of reliable speech recognition systems for speaker recognition. High-level features not only offer the potential to improve accuracy, they may also help improve robustness since they should be less affected by channel effects.

Both speech and speaker recognition rely primarily on spectral features, but speaker recognition has made more use of prosodics, such as the pitch, timber, intonation, and so forth. Recent work has shown that such levels of information can indeed be exploited and used profitably in automatic speaker recognition systems [32]. On the other hand, these features can be difficult to extract by machine form the speech signal itself, for automatic speaker recognition.

2.6 System Design

Designing a speaker identification or verification system needs to take into consideration many parameters. The characteristics of speakers are quite difficult to describe, compared to other systems such as speech recognition systems. Furthermore, comparing identification experiments using different sets of speakers is difficult, since one study may use a

homogeneous set of speakers e.g. one gender, narrow age range, persons raised in a small geographical area, while another uses a heterogeneous set e.g. males and females of varying ages and dialect. The latter yields much higher recognition accuracy. As a result, the utility of many speaker identification studies is limited and some systems can be quite subjective.

2.6.1 Data Collection

Besides speaker selection, the time span over which training speech is collected is of crucial importance for identification performance. Speaking style often changes substantially during the course of the day, from day to day, and over longer periods of time. Experiments using reference and test data from the same recording session usually yield high recognition accuracy (e.g. using the TIMIT database), which is misleading since practical applications often "compare" test data with reference data that were obtained much earlier. Performance usually decreases as the interval between training and testing increases. The cure to this problem is that reference data must be updated periodically. Some systems consider each recognized utterance as new training data, revising the reference models to reflect changes in a person's speaking style over time. Generally, in any pattern recognition task, training and testing data should be kept separate. If the same utterances are used to train and test a recognizer, artificially high accuracy follows. With common training and testing data the system possibly may not be reliable for new test data. For example, with k utterances per speaker as data, a common procedure trains the system using $k-1$ as training data and one as test, but repeats the process k times treating each utterance as test once. Technically, this is called *leave-one-out* method. This method verifies whether

the system is good using limited amount of data, while avoiding the problem of common training and testing speech.

2.6.2 Multi-stage Recognition

In speaker identification, computation and response time usually increases linearly with population size (i.e the number of speakers whose models have been stored) because each speaker's model must be examined. One way to minimize computation is to set up a hierarchy so that speakers are clustered into groups that can be easily and rapidly identified from a test utterance. A simplified example is to classify speakers according their gender at the first stage, in this way you decrease the population in half. This is quite similar to the idea of *cohorts* where each speaker has similar impostors (i.e. speakers acoustically close to them) to reduce redundancy in the background speaker set. Pooled models based on speaker similarity seems to be better than individual cohort models [33]. The multi-stage method may also help in recognizing speakers with varying dialects. If speakers can be partitioned by dialect, certain prosodic features have been shown to be effective in distinguishing them.

One measure that estimates speaker similarity is a log likelihood distance, as shown in the following formula:

$$d(\lambda_A, \lambda_B) = \log \frac{p(X_A/\lambda_A)}{p(X_A/\lambda_B)} + \log \frac{p(X_B/\lambda_B)}{p(X_B/\lambda_A)} \tag{2.5}$$

where X_A and X_B are feature vectors (or sequences of feature vectors) from two speakers A and B, respectively, and λ_A and λ_B are their speaker models [34].

2.6.3 Effects of Different Communication Channels

Since many speaker verification applications involve telephone speech or speech being subjected to other environmental distortion, the effects on recognition accuracy due to environment must be considered carefully. As we will see, cepstral coefficients have the advantage of being invariant under linear distortions. Among the distortions that a telephone link introduces is bandpass filtering. Successful speaker verification on such limited speech have been the use of cepstral coefficients.

Telephone distortions introduces serious difficulties for speaker identification or verification. A large percentage of calls for customer service contain speech distorted by music, traffic noise etc. Another major cause of mismatch between training and testing speech is the type of handset used. One study found large performance improvement with a handset normalization procedure [34].

Because different links demonstrate large variability in quality compared with that during transmission over one link, it is important for telephone speaker verification to train and test using different links. One study, using ten speakers and cepstral features, found that the error rate increased from 17% to 56% when the test data were drawn from a different link than the training data [24]. Normalizing each feature over a recording session, by subtracting off its mean, to produce channel-invariant features reduces this degradation but also decreases overall performance because useful speaker information is also eliminated.

2.7 Speaker Recognition by Humans

People can reliably identify voices, although error rates often exceed 20% for brief utterances. About $2 - 3$ *sec* of speech is sufficient to identify a voice, although performance decreases for unfamiliar voices. Speaker recognition is one area of artificial intelligence where machine performance can be superior to human performance. Especially, using short test utterances and a large number of speakers, speaker identification accuracy by machine often exceeds that of humans. This is particularly true for unfamiliar speakers, where the training time for humans to learn a new voice well is very long compared with that for machines. Constraints on how many unfamiliar voices a person can retain in their memory usually limit studies of speaker recognition by humans to about $5 - 10$ speakers. Such small speaker sets lead to large statistical variations from one set to another because the ability to distinguish and the degree of familiarity of voices often varies widely across different speakers.

How do humans recognize the voices of persons is not completely known by researchers. Generally, they use multiple levels of speaker information conveyed in the speech signal. At the lowest level, they recognize a person based on the sound of their voice (e.g. low/high pitch, bass/tenor, nasality, etc.). But we also use other types of information in the speech signal to recognize a speaker, such as a unique laugh, particular phrase usage, or the speed of speech, among other things. Most current state-of-the-art automatic speaker recognition systems, however, use only low-level sound information (specifically, features based on purely acoustic cues computed on $10 - 20$ *ms* intervals of speech) and ignore higher-level information. While these systems have shown reasonably high performance, there is much

more information in the speech signal that can be used and, potentially, greatly improve accuracy and robustness.

Chapter 3

Speaker Recognition System Design

3.1 Introduction

Speaker identification and verification system design relies heavily on pattern recognition, one of the most challenging problems for machines. In a broader sense, the ability to recognize patterns forms the core of our intelligence. If we can incorporate the ability to reliably recognize patterns in our work and life, we can make machines much easier to use. Up to now, the process of human pattern recognition is not well understood.

Due to the inherent variability of the speech signal, as far as the identity of a person is concerned, we emphasize the use of statistical approaches in this thesis. The decision of pattern recognition is based on appropriate probabilistic models of the patterns. This chapter presents several mathematical fundamentals for statistical pattern recognition and classification. In particular, Bayes' decision theory and esimation techniques for parameters of classifiers are introduced. Bayes' decision theory, which plays a central role for statistical pattern recognition, introduces the concept of decision-making based on both posterior knowledge obtained from specific observation data, and prior knowledge based of the various categories. To build such a classifier, it is crucial to estimate prior class probabilities and the class-conditional probability densities for a Bayes' classifier.

The technique we use goes under the category of supervised learning, as long as class information is available for the data. Only the probabilistic structure needs to be learned. Maximum likelihood estimation is a powerful method, that we use. The Expectation Maximization (EM) algorithm is an iterative procedure for our model parameter estimation. The EM algorithm forms the theoretical basis for training Hidden Markov Models (HMM).

This chapter also describes the form of the Gaussian mixture model (GMM) and motivates its use as a representation of speakers for text-independent speaker identification and verification. The speech analysis for extracting the mel-cepstral feature representation used in this work is presented first. Next, the Gaussian mixture speaker model and its parameterization are described. The use of the Gaussian mixture density for speaker identification is then described. Finally, the maximum-likelihood parameter estimation and speaker identification procedures are presented.

3.2 Bayes' Theorem in General

For continuous variables the prior probabilities can be combined with the class-conditional densities to give the posterior probabilities $P(C_k|x)$ using the well known Bayes' formula. For c different classes $C_1, \ldots, C_c$ and for a continuous feature vector x, we can write the Bayes' theorem in the form

$$P(C_k|x) = \frac{p(x|C_k)P(C_k)}{p(x)} \tag{3.1}$$

where $p(x)$ is the unconditional density function, that is the density function for x irrespective of the class, and is given by

$$p(x) = \sum_{k=1}^{c} p(x|C_k)P(C_k) \tag{3.2}$$

which ensures that the posterior probabilities sum to unity

$$\sum_{k=1}^{c} P(C_k) = 1 \tag{3.3}$$

In practice, we might choose to model the class-conditional densities $p(x|C_k)$ by parameterized functional forms, as we will see later in this text. When viewed as functions of the parameters they are referred to as likelihood functions, for the observed value of x. Bayes' theorem can therefore be summarized in the form

$$posterior = \frac{likelihood \times prior}{normalization\ factor} \tag{3.4}$$

3.3 Probability Density

The problem of modeling a probability density function $p(x)$, given a finite number of data x_n, $n = 1, 2 \cdots N$, drawn from the real density function, is of fundamental consideration in every pattern recognition problem, like the problem of speaker identification. The method we describe can be used to build a classifier system by considering each of the classes C_k in turn, and estimating the corresponding conditional densities $p(x|C_k)$ by making use of the fact that each data point is labeled according to its class. These densities can then be used in Bayes' theorem to find the posterior probabilities corresponding to a new measurement of x, which can in turn be used to make a classification of x.

Density estimation can be applied to unlabeled data (that is data without any class labels) where it has a number of applications. In the context of neural networks it can be applied to the distribution of data in the input space as part of the training process for radial basis function networks, and to provide a method for validating the outputs of a trained neural network.

There are three alternative approaches to density estimation. The first of three involves parametric methods in which a specific form of the density model is assumed. This contains a number of parameters which are then optimized (optimally estimated) by fitting the model to the data set. The drawback of such an approach is that the particular form of parametric function chosen might be incapable of providing a good representation of the density. By contrast, techniques of non-parametric estimation do not assume a particular functional form, but allow the form of the density to be determined entirely by the data. Another approach, sometimes called semi-parametric estimation, tries to achieve the best of both techniques by allowing a very general class of functional forms in which the number of adaptive parameters can be increased in a systematic way to build ever more flexible models, but where the total number of parameters in the model can be varied independently from the size of the data set. In this thesis, we will focus on semi-parametric models based on *mixture distributions*.

3.3.1 Parametric Techniques

One of the most straightforward approaches to density estimation is to represent the probability density $p(x)$ in terms of a specific functional form which contains a number of

adjustable parameters. The values of the parameters can then be optimized to give the best fit to the data. The simplest and most widely used parametric model is the normal or Gaussian distribution, which has a number of convenient analytical and statistical properties.

In d dimensions the general multivariate normal probability density can be written

$$p(x) = \frac{1}{(2\pi)^{d/2}|\Sigma|^{1/2}} \exp\left\{-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)\right\} \tag{3.5}$$

where the mean μ and x are now d-dimensional vectors, Σ is a $d \times d$ covariance matrix, and $|\Sigma|$ is the determinant of Σ. The factor in Equation (3.5) ensures that $\int_{-\infty}^{\infty} p(x)dx = 1$, as can again be verified. The density function $p(x)$ is governed by the parameters μ and Σ, which satisfy

$$\mu = E[x] \tag{3.6}$$

$$\Sigma = E[(x-\mu)(x-\mu)^T] \tag{3.7}$$

Observe that Σ is a symmetric matrix, and therefore has $d(d+1)/2$ different values. There are also d different elements in μ, and so the density function is completely specified once the values of $d(d+3)/2$ parameters have been determined. A surface plot of the normal distribution for the case of two dimensions (i.e. $d = 2$) is shown in Figure 3.1.

3.3.2 Maximum Likelihood

Having decided for the parametric form for a density function $p(x)$, the next stage is to use the data set to find values for the parameters. The approach we describe here is known as maximum likelihood. Maximum likelihood seeks to find the optimum values for the parameters by maximizing a likelihood function derived from the training data.

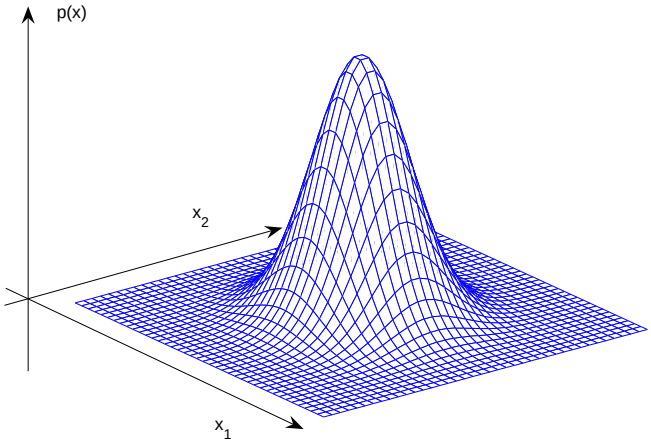


Figure 3.1: Surface plot representing the Gaussian probability density for the case of two dimensional feature vectors.

Suppose we consider a density function $p(x)$ which depends on a set of parameters $\theta = (\theta_1, \theta_2 \cdots \theta_M)^T$. In a classification problem we would take one such function for each of the classes. Here we will omit the class labels for simplicity, but essentially the same steps are performed separately for each class in the problem. To make the dependence on the parameters explicit, we will write the density function in the form $p(x|\theta)$. We also have a data set of N vectors $X = \{x_1, x_2 \cdots x_N\}$. If these vectors are drawn independently from the distribution $p(x|\theta)$, then the joint probability density of the whole data set X is given by

$$p(X|\theta) = \prod_{n=1}^{N} p(x_n|\theta) \equiv \mathcal{L}(\theta) \tag{3.8}$$

where $\mathcal{L}(\theta)$ can be viewed as a function of θ for fixed X, in which case it is referred to as the likelihood of θ for the given X. The technique of maximizing likelihood then sets the value of θ by maximizing $\mathcal{L}(\theta)$. This corresponds to the intuitively reasonable idea of

choosing the θ which is most likely to cause the observed data. In practice, it is often more convenient to consider the negative logarithm of the likelihood

$$E = -\ln \mathcal{L}(\theta) = -\sum_{n=1}^{N} \ln p(x_n|\theta) \tag{3.9}$$

and to find the minimum of E. This is equivalent to maximizing $\mathcal{L}$ since the negative logarithm is monotonically decreasing function. The negative log-likelihood can be regarded as an error function which we want to minimize.

For most choices of density function, the optimum θ will have to be found found by an iterative numerical procedure. However, for the special case of multivariate normal density, we can find the maximum likelihood solution by analytic differentiation of Equation (3.9), with $p(x|\theta)$ given by Equation (3.5). With some straightforward but rather involved matrix algebra, the above calculations lead to the following results

$$\hat{\mu} = \frac{1}{N}\sum_{n=1}^{N} x_n \tag{3.10}$$

$$\hat{\Sigma} = \frac{1}{N}\sum_{n=1}^{N} (x_n - \hat{\mu})(x_n - \hat{\mu})^T \tag{3.11}$$

which represents the intuitive result that the maximum likelihood estimate $\hat{\mu}$ of the mean vector μ is given by the sample average (i.e. the average with respect to the given data set). Similarly, the maximum likelihood estimate $\hat{\Sigma}$ of the covariance matrix Σ is given by the sample average of the matrices $(x_n - \hat{\mu})(x_n - \hat{\mu})^T$.

3.4 Mixture Models

There are two general approaches to density estimation, parametric and non-parametric, each of which has its merits and limitations. In particular, the parametric approach assumes

a specific form for the density function, which might be very different from the true density. Usually, however, parametric models allow the density function to be evaluated very rapidly for new values of the input vector. Non-parametric methods, by contrast, allow very general forms of density function, but suffer from the fact that the model can be very slow to evaluate for new input vectors.

In order to combine the advantages of both parametric and non-parametric methods we need to find techniques which are not restricted to specific functional forms, and the size of the model only grows with complexity of the problem being considered, and not with the size of the data set. This leads to a class of models which are generally known as *semi-parametric*. The price we have to pay is that the training of the model using the data set is computationally intensive compared to the simple procedure needed for parametric methods which in some cases involve little more than evaluating a few expressions for parameter values.

In this thesis we will restrict attention to one particular form of density function, called a *mixture model*. As well as providing powerful techniques for density estimation (see EM algorithm). Mixture models find important applications in the context speaker recognition. We will also discuss a training method for mixture models, which is based on maximum likelihood principle, involving re-estimations leading to the well known Expectation Maximization (EM) algorithm.

In the non-parametric kernel based approaches to density estimation, the density function was represented as a linear superposition of kernel functions (e.g. Gaussian functions), with one kernel centered on each data point. Here we consider models in which the density function is again formed from linear combination of basis functions, but where the number

M of basis functions is treated as parameter of the model and is typically much less than the number N of the data points. We therefore write our model for the density as a linear combination of component densities $p(x|j)$ in the form

$$p(x) = \sum_{j=1}^{M} p(x|j)P(j) \tag{3.12}$$

Such a representation is called as a *mixture distribution.* The coefficients $P(j)$ are called the mixing parameters or weights. Notice that there is a strong similarity between Equation (3.12) and the expression given in Equation (3.2) for the unconditional density of data taken from a mixture of several classes. This similarity has been emphasized by our choice of notation. We will call $P(j)$ the *prior* probability of the data point having been generated from component j of the mixture. These priors are chosen to satisfy the constraints

$$\sum_{j=1}^{M} P(j) = 1 \tag{3.13}$$

$$0 \leq P(j) \leq 1 \tag{3.14}$$

Similarly, the component density functions $p(x|j)$ are normalized so that

$$\int_{-\infty}^{\infty} p(x|j)dx = 1 \tag{3.15}$$

and hence can be regarded as class-conditional densities. To generate a data point from the probability distribution shown in Equation (3.12), one of the components j is first selected at random with probability $P(j)$, and then a data point is generated from the corresponding component density $p(x|j)$. An important property of such mixture models is that, for many choices of component density function, they can approximate any continuous density to arbitrary accuracy provided the model has sufficiently large number of components, and provided the parameters of the model are chosen correctly.

The important difference between the mixture model representation and the true classification problem lies in the nature of the training data, since in this case we are not provided with any class labels to say which component was responsible for generating each data point. This represents an example of incomplete data. As with any other density, the mixture modeling can be applied separately to each class C_k in a classification problem, as the one we consider in this work. In this case, each class-conditional density $p(x|C_k)$ is represented by an independent mixture model of the form of Equation (3.12).

Having made the link with prior probabilities and conditional densities, we can introduce the corresponding posterior probabilities, which we can express using Bayes' theorem in the form

$$P(j|x) = \frac{p(x|j)P(j)}{p(x)} \tag{3.16}$$

where $p(x)$ is given by Equation (3.12). These posterior probabilities satisfy

$$\sum_{j=1}^{M} P(j|x) = 1 \tag{3.17}$$

The value of $P(j|x)$ represents the probability that the particular component j was responsible for generating the data point x. In this thesis, we will limit our attention to mixture models in which the individual component densities are given by Gaussian distribution functions as given in Equation (3.5). Figure 3.2 shows a mixture of Gaussian density functions for the cases of one dimension.

3.5 Estimation via Maximum Likelihood

Various procedures have been developed for determining the parameters of a Gaussian mixture model from a set of data. In the following we will consider an approach based on

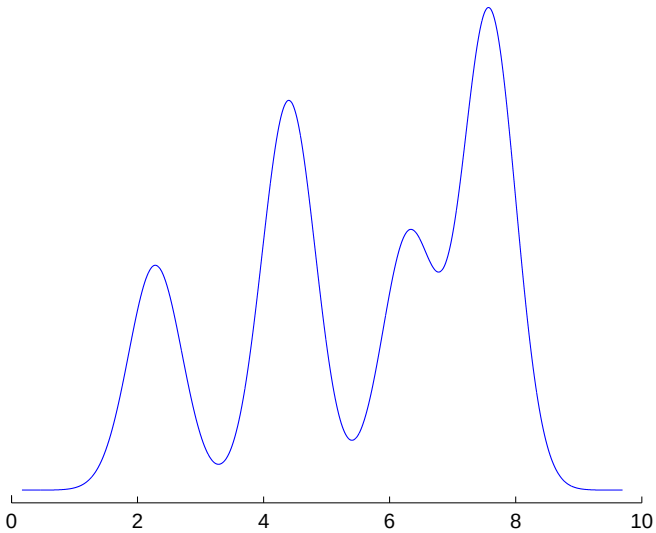


Figure 3.2: Curve representing a probability density function composed of a mixture of Gaussian functions.

maximizing the likelihood of the parameters for the given data set.

For the case of Gaussian components of the form as shown by Equation (3.5), the mixture density contains the following adjustable parameters: $P(j)$, μ_j, and Σ_j where $j = 1, \ldots, M$. The negative log-likelihood for the data set is given by

$$E = -\log \mathcal{L} = -\sum_{n=1}^{N} \log p(x_n) = -\sum_{n=1}^{N} \log \left\{ \sum_{j=1}^{M} p(x_n|j)P(j) \right\} \tag{3.18}$$

which can be regarded as an error function. Maximizing the likelihood $\mathcal{L}$ is then equivalent to minimizing E.

It is important to emphasize that minimizing this error function in a non-trivial procedure. First of all, there exist parameter values for which the likelihood goes to infinity. This case arises when the Gaussian components collapse onto one of the data points. In

addition, small groups of points which are close together can give rise to local minima in the error function which may give poor representations of the true distribution. In practical problems we wish to avoid the singular solutions and the inappropriate local minima. Several techniques for dealing with the problems of singularities have been proposed.

Since the likelihood function is a smooth differentiable function of the parameters of the mixture model, we can employ standard non-linear optimization techniques to find the maxima. A necessary condition for the maximum likelihood parameter estimation, is that they satisfy the following equation

$$\frac{\partial p(X|\theta)}{\partial \theta} = 0 \tag{3.19}$$

Unfortunately, attempting to solve Equation (3.19) directly for the mixture parameters, does not yield closed form solutions.

The maximum likelihood parameter estimation of mixture models can be accomplished via an iterative parameter estimation procedure, which is known as *Expectation Maximization* (EM) algorithm. The EM is the basis for several parameter estimation procedures in the area of statistical data analysis. The widespread use of the EM algorithm stems from the facts that it guarantees a non-decreasing likelihood function after each iteration and that it provides a general yet powerful framework capable of dealing with many complicated estimation problems. The basis idea of the EM algorithm is, beginning with an initial model θ, to estimate a new model $\bar{\theta}$, such that $p(X|\bar{\theta}) \geq p(X|\theta)$. The new model then becomes the current model and the process is repeated until some convergence threshold is reached. Figure 3.3 shows four typical examples of the behavior of the log likelihood function for several iterations of the EM algorithm when training a Gaussian mixture model for

the speaker identification system described in this thesis. Note that after an initial large increase in the likelihood function, each successive iteration only increases the likelihood function slightly, which means that the algorithm have converged to a density for the given data set.

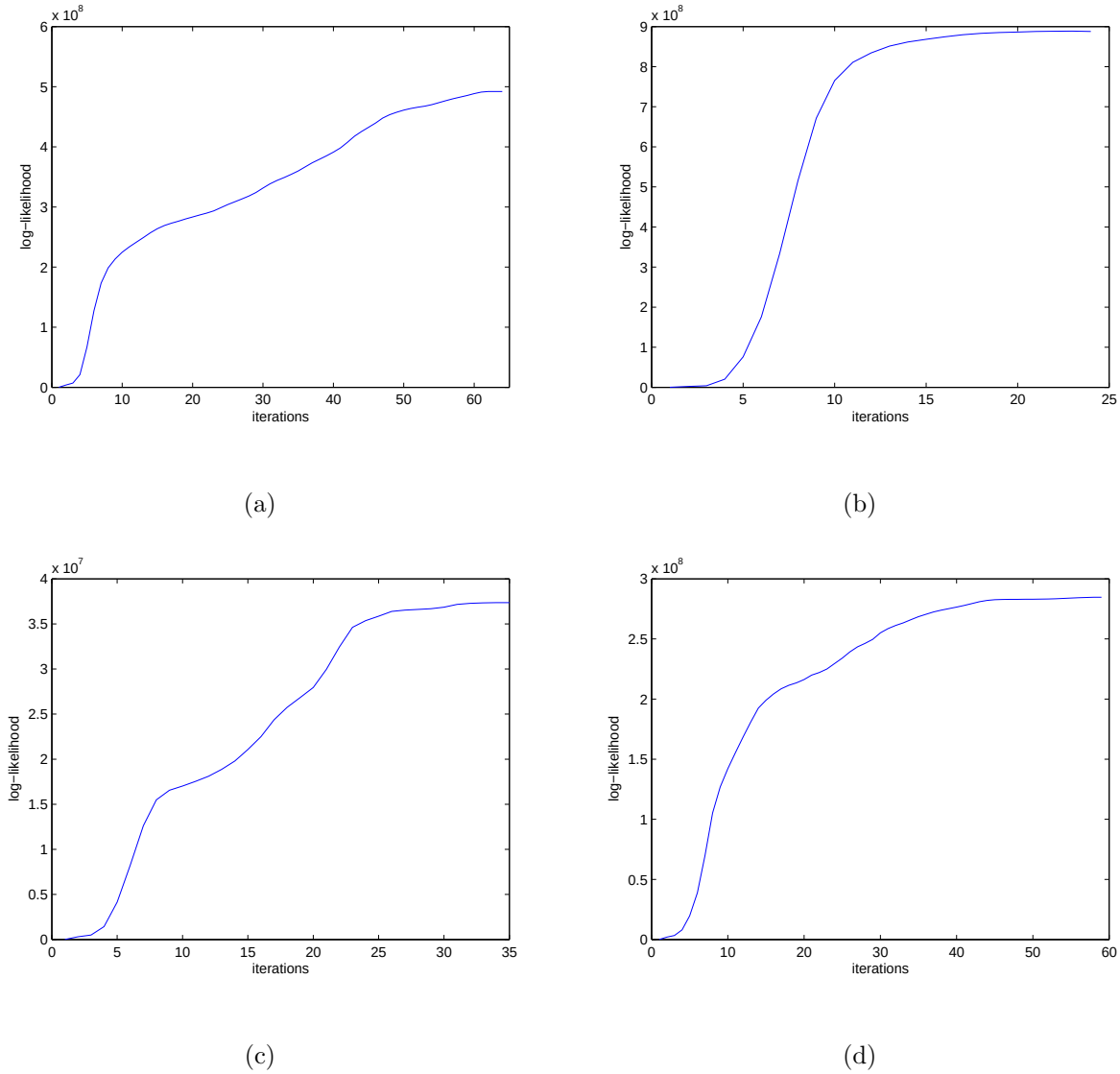


Figure 3.3: Log-likelihood behavior for four different speaker models in the speaker recognition system.

3.5.1 Multivariate Gaussian Mixture Density Estimation

In this section we describe the maximum-likelihood parameter estimation problem and how the Expectation- Maximization (EM) algorithm can be used for its solution. We develop the EM parameter estimation procedure for the application of finding the parameters of a mixture of Gaussian densities. We try to emphasize intuition rather than mathematical details.

Recall the definition of the maximum-likelihood estimation problem. We have a density function $p(x|\theta)$ that is governed by the set of parameters θ (e.g. $p(x)$ might be a mixture of Gaussians and θ could be the means and covariances). We also have a data set of size N, supposedly drawn from this distribution, i.e. $X = \{x_1, \ldots, x_N\}$. That is, we assume that these data vectors are independent and identically distributed (i.i.d.) with distribution p. Therefore, the resulting density for the samples is

$$p(X|\theta) = \prod_{i=1}^{N} p(x_i|\theta) = \mathcal{L}(\theta|X) \tag{3.20}$$

The function $\mathcal{L}(\theta|X)$ is called the likelihood of the parameters given the data, or just the likelihood function as we saw previously in this chapter. The likelihood is thought of as a function of the parameters θ where the data X is fixed. In the maximum likelihood problem, our goal is to find the θ that maximizes $\mathcal{L}$. That is, we wish to find θ^* where

$$\theta^* = \arg\max_{\theta} \mathcal{L}(\theta|X) \tag{3.21}$$

Often we maximize $\log(\mathcal{L}(\theta|X))$ instead because it is analytically easier.

Depending on the form of $p(x|\theta)$ this problem can be easy or hard. For example, if $p(x|\theta)$ is simply a single Gaussian distribution where $\theta = (\mu, \sigma^2)$, then we can set the

derivative of $\log(\mathcal{L}(\theta|X))$ to zero, and solve directly for μ and σ^2 (this, in fact, results in the standard formulas, for the mean and variance of a data set, that we saw previously in this chapter). For many problems, however, it is not possible to find such analytical expressions, and we must resort to more elaborate techniques.

The EM algorithm is one such elaborate technique. The EM algorithm [35–37] is a general method of finding the maximum-likelihood estimate of the parameters of an underlying distribution from a given data set.

Therefore, based on the EM algorithm we obtain the following re-estimation formulas for a_j, μ_j, and Σ_j

$$P(j)^{\text{new}} = \frac{1}{N}\sum_{n=1}^{N} P^{\text{old}}(j|x_n) = \frac{1}{N}\sum_{n=1}^{N} \frac{p(x_n|j)P(j)}{p(x_n)} \tag{3.22}$$

$$\mu_j^{\text{new}} = \frac{\sum_{n=1}^{N} P^{\text{old}}(j|x_n)x_n}{\sum_{n=1}^{N} P^{\text{old}}(j|x_n)} \tag{3.23}$$

$$\Sigma_j^{\text{new}} = \frac{\sum_{n=1}^{N} P^{\text{old}}(j|x_n)(x_n-\mu_j^{\text{new}})(x_n-\mu_j^{\text{new}})^T}{\sum_{n=1}^{N} P^{\text{old}}(j|x_n)} \tag{3.24}$$

The quantity $P(j)$ represents the weights a_j and $p(x_n) = \sum_{j=1}^{M} p(x_n|j)P(j)$ where M is the number of Gaussian components.

3.6 Speech Analysis

This section describes how to extract speech signal properties or features important for recognition, from a speech signal $x(n)$, a process called speech analysis. This involves the transformation of $x(n)$ into another signal or a set of parameters with the objective of simplification or data reduction. The relevant information in speech for different applications can often be expressed very compactly. In speech analysis we wish to extract features

directly pertinent for speaker recognition, while suppressing redundant aspects of speech. The original speech signal may approach optimality from the point of view of human perception, but it has much repetitive data when processed by a computer. Eliminating such a redundancy aids accuracy in computer applications, such as speaker recognition.

Eliminating redundancy and irrelevant aspects of the speech waveform simplifies data manipulation. An efficient representation for speaker recognition would be a set of parameters consistent across speakers, yielding similar values for the same speaker in different sessions, while exhibiting reliable variation for different speakers.

In the following we investigate methods of speech analysis. We want to obtain a more useful representation of the speech signal in terms of parameters that contain relevant information in an efficient format. We will describe the issues involved in analyzing speech as a time-varying signal. Analyzers periodically examine a limited time range (i.e. a window) of the speech signal. The choice of duration and shape for the window reflects a compromise in time and frequency resolution.

3.6.1 Short-time Speech Analysis

Speech is dynamic or time-varying, some variation is under speaker control, but much is random. Aspects of the speech signal directly characterize a speaker and methods to extract related parameters from the speech signal is the primary interest here.

During slow speech, the vocal tract shape and type of excitation may not alter for durations up to 200 *ms*. Mostly, however, they change more rapidly since phoneme average about 80 *ms*. Nonetheless, speech analysis usually assumes that the signal properties change

relatively slowly with time. This allows examination of a short-time window of speech to extract parameters presumed to remain fixed for the duration of the window. Thus, to model dynamic parameters, we must divide the signal into successive windows or analysis frames, so that the parameters can be calculated often enough to follow relevant changes due to dynamic vocal tract configurations. Slowly changing parameters in long vowels may allow windows as large as 100 ms without obscuring the desired parameters via averaging, but rapid events require short windows of about $5 - 10$ ms to avoid averaging spectral transitions. In speaker recognition system we have chosen to use 20 ms windows as a trade-off between the two extremes.

3.6.2 Windowing

Windowing is multiplication of a speech signal $x(n)$ by a window $w(n)$, which yields a set of speech samples $s(n)$ weighted by the shape of the window. $w(n)$ may have infinite duration, but in our case has finite length to simplify computation. By shifting $w(n)$, we examine any part of $x(n)$ through the movable window.

Many applications prefer some speech averaging, to yield an output parameter contour that represents some slowly varying physiological aspects of vocal tract movements. The amount of the desired smoothing leads to a choice of window size trading off three factors. Firtly, $w(n)$ short enough that the speech properties of interest change little within the window. Secondly, $w(n)$ long enough to allow calculating the desired parameters e.g. if additive noise is present, longer windows can average out some of the random noise. Thirdly, successive windows not so short as to omit sections of $x(n)$ as an analysis is periodically

repeated. The last condition reflects more on the frame rate , that is, the number of times per second that speech analysis is performed, advancing the window periodically in time, than on window size. Normally, the frame rate is about twice the inverse of the duration of $w(n)$, so that successive windows overlap, which is important in the common case that $w(n)$ has a shape that de-emphasizes speech samples near its edges. In the speaker recognition system we have used frame rate of windows that overlap by 50%, as we will see later in this text.

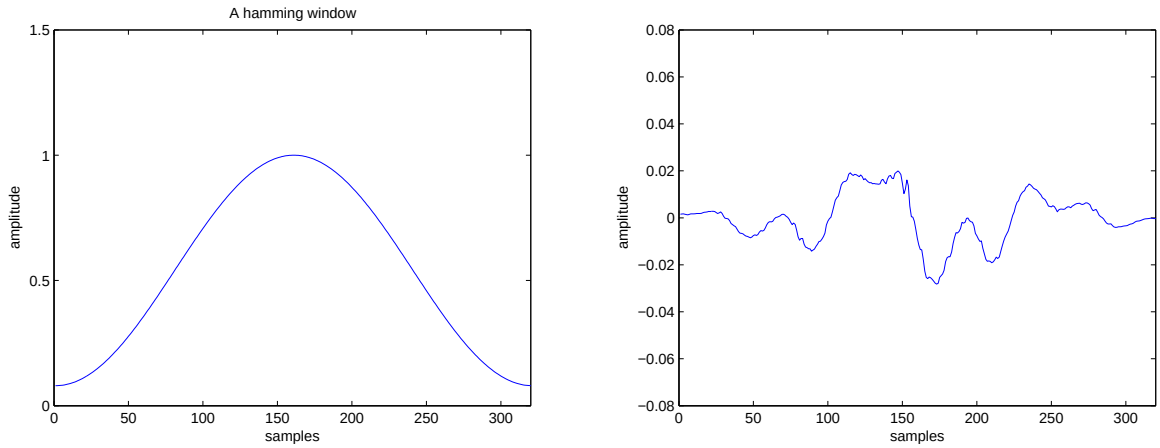
The size and shape of $w(n)$ depends on their effects on speech analysis. Typically $w(n)$ is smooth, because its values determine the weighting of $x(n)$ and a priori all samples are equally relevant. Except at its edges, $w(n)$ rarely has sudden changes. In particular, windows rarely contain zeros or negative-valued points. The simplest window has a rectangular shape

$$w(n) = r(n) = \begin{cases} 1 & 0 \leq n \leq N-1 \\ 0 & \text{otherwise} \end{cases} \tag{3.25}$$

This choice provides equal weight for all samples, and just limits the analysis range to N consecutive samples. However, weighting the middle samples more than the edge, relates to the effect that window shape has on the output speech parameters. When $w(n)$ is shifted to analyze successive frames of $x(n)$, large changes in output parameters can arise when using rectangular shaped window. A common alternative to Equation (3.25) is the Hamming window, a raised cosine pulse, given in the following formula

$$w(n) = h(n) = \begin{cases} 0.54 - 0.46\cos(\frac{2\pi n}{N-1}) & 0 \leq n \leq N-1 \\ 0 & \text{otherwise} \end{cases} \tag{3.26}$$

Tapering the edges of $w(n)$ allows its periodic shifting along $x(n)$ without having effects on the speech parameters. This window have been used for speech analysis in the speaker recognition system. Figure 3.4 shows a typical Hamming window of 20 ms and a speech segment multiplied by the Hamming window to produce a short-time frame for analysis.



(a) (b)

Figure 3.4: (a) Hamming window of 20 ms duration sampled by a sampling frequency of 16 kHz. (b) A speech frame of 20 ms for analysis.

3.6.3 Frequency-domain Parameters

The frequency domain provides most useful parameters for speech processing. Speech signals are more consistently and easily analyzed spectrally than in the time domain. The basic model of speech production with a noisy or periodic waveform that excites a vocal tract filter corresponds well to separate spectral models for the excitation and for the vocal tract. Repeated utterances of a sentence by a speaker often differ greatly temporally while being very similar spectrally. Human hearing appears to pay much more attention

to spectral aspects of speech (e.g. amplitude distribution in frequency) than to phase or timing aspects. Thus, spectral analysis is used to extract parameters from speech for speaker identification and verification.

3.6.4 Filterbank Analysis

One spectral analysis method, popular due to real-time and inexpensive implementation, uses a filter bank or set of bandpass filters, each analyzing a different range of frequencies of the input speech. Filter banks are more flexible than DFT analysis since the bandwidths can be varied to follow the resolving power of the ear, rather than being fixed, as in DFT. Furthermore, many applications require a small set of parameters describing the spectral distribution of energy, especially the spectral envelop. The amplitude outputs from a bank of $10 - 20$ bandpass filters provide a more efficient spectral representation than a more detailed DFT. Filters often follow the bark scale or mel scale, i.e. equally spaced , fixed-bandwidth filters up to 1 kHz, then logarithmically increasing bandwidth. In the speaker recognition system, we have used both scales as it is described later in this thesis.

3.6.5 Short-time Fourier Transform Analysis

As the traditional spectral techniques, Fourier analysis provides a speech representation in terms of amplitude and phase as a function of frequency. Viewing the vocal tract as a linear system, the Fourier transform of speech is the product of the transforms of the glottal excitation and the vocal tract response. For steady-state vowels or fricatives, the basic (infinite-time) Fourier transform could be used by extending or repeating sections or

pitch periods of the speech without limit. However, speech is not a stationary signal, and thus short-time analysis using windows is necessary.

The short-time Fourier transform of the signal $s(n)$ is often defined as

$$S_n(e^{j\omega}) = \sum_{m=-\infty}^{\infty} s(m)e^{j\omega}w(n-m) \tag{3.27}$$

If ω is considered fixed , $S_n(e^{j\omega})$ is a time signal (a function of n), describing the amplitude and phase of $s(n)$ within a bandwidth equivalent to that of the window but centered at the fixed frequency ω. Repeating the calculation of $S_n(e^{j\omega})$ at different ω of interest yields a two-dimensional representation of the input speech, i.e. an array of time signals indexed on frequency, each noting the speech energy in a limited bandwidth about the chosen frequency.

A second interpretation of $S_n(e^{j\omega})$ views n as fixed, thus yielding the Fourier transform of the speech segment $s(m)w(n-m)$, the windowed version of $s(m)$ using a window shifted to a time n with respect to the speech. This calculation can be repeated for successive n to produce an array of Fourier transforms indexed on time n, each expressing the spectrum of the speech signal within a window centered at time n.

For computational purposes, the DFT is used instead of the standard Fourier transform, so that the frequency variable ω only takes on N discrete values, that is

$$S_n(k) = \sum_{m=0}^{N-1} s(m)e^{-j2\pi km/N}w(n-m) \tag{3.28}$$

In practice, each frame of speech samples $s(m)$ is shifted by the time delay n to align with a start at $m = 0$, allowing a simple N-sample window $w(m)$ to replace $w(n-m)$, and the fast Fourier transform or FFT is used to implement the DFT.The choice of N is crucial for short-time Fourier analysis. Low values of N (i.e. short windows) give poor frequency

resolution since the window lowpass filter is wide, but yield good time resolution since the speech properties are averaged over short time intervals. Large N, on the other hand, give poor time resolution and good frequency resolution.

3.7 Cepstral Coefficients

The most popular analysis method for automatic speaker recognition uses the cepstrum, with nonlinear frequency axis following the Bark or mel scale. Table 3.1 shows the Bark scale ranging from 1 to 21 Barks, corresponding to 21 critical bands of hearing. Such mel-frequency cepstal coefficients (MFCC) provide an alternative representation for speech spectra which incorporates some aspects of audition. The extraction of such reliable features is one of the most important issues in speaker identification and verification. There are also a large number of features we can use, besides. However, the *curse-of-dimensionality* problem reminds us that the amount of training data is always limited. Therefore, incorporation of additional features may not lead to any measurable error reduction. This does not necessarily mean that the additional features are poor ones, but rather that we may have insufficient data to reliably model those features.

The first feature we can use is the speech waveform itself. In general, time-domain features are much less accurate than frequency-domain features such as the mel-frequency ceptral coefficients (MFCC). When 16 *kHz* sampling rate is used, a typical state-of-the-art speaker recognition system can be build based on 20th-order MFCC c_k. The initial coefficient c_0 represents the average energy in the speech frame and is often discarded as amplitude normalization. c_1 reflects the energy balance between low and high frequencies.

Table 3.1: The Bark frequency scale.

Bark Band #	**Edge (Hz)**	**Center (Hz)**
1	100	50
2	200	150
3	300	250
4	400	350
5	510	450
6	630	570
7	770	700
8	920	840
9	1080	1000
10	1270	1170
11	1480	1370
12	1720	1600
13	2000	1850
14	2320	2150
15	2700	2500
16	3150	2900
17	3700	3400
18	4400	4000
19	5300	4800
20	6400	5800

A short-time analysis Hamming window of $20ms$ is typically used to compute the MFCC vector c_k. The window shift is typically $10ms$, as we will see later in this text in more detail.

3.7.1 Mel-frequency Ceptrum

The Mel-frequency Cepstral Coefficients (MFCC) is a representation defined as the real cepstrum of a windowed short-time signal derived from the FFT of that signal. The difference from the real cepstrum is that a nonlinear frequency scale is used , which approximates the behavior of the auditory system. Davis and Mermelstain [17] helped us to understand that MFCC representation is beneficial for speaker recognition.

Given the DFT of the input signal

$$X[k] = \sum_{n=0}^{N-1} x[n]e^{-j2\pi nk/N}, \quad 0 \leq k < N \tag{3.29}$$

we define a filterbank with M filters ($m = 1, \ldots, M$), where filter m is a triangular filter given by

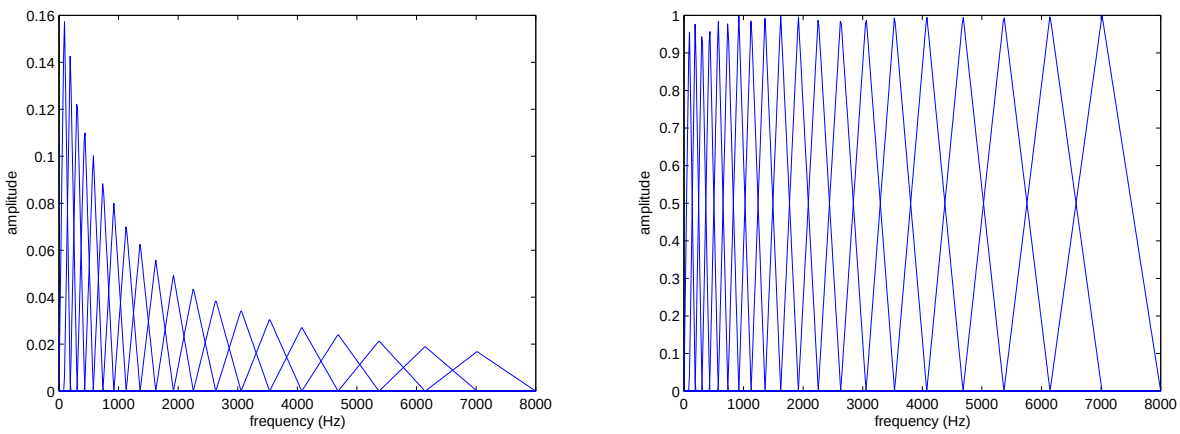
$$H_m[k] = \begin{cases} 0 & k < f[m-1] \\ \frac{2(k-f[m-1])}{(f[m+1]-f[m-1])(f[m]-f[m-1])} & f[m-1] \leq k \leq f[m] \\ \frac{2(f[m+1]-k)}{(f[m+1]-f[m-1])(f[m+1]-f[m])} & f[m] \leq k \leq f[m+1] \\ 0 & k > f[m+1] \end{cases} \tag{3.30}$$

Such filters compute the average spectrum around each center frequency with increasing bandwidth.

Alternatively, the filters can be chosen as

$$\bar{H}_m[k] = \begin{cases} 0 & k < f[m-1] \\ \frac{(k-f[m-1])}{(f[m]-f[m-1])} & f[m-1] \leq k \leq f[m] \\ \frac{(f[m+1]-k)}{(f[m+1]-f[m])} & f[m] \leq k \leq f[m+1] \\ 0 & k > f[m+1] \end{cases} \tag{3.31}$$

which satisfies $\sum_{k=0}^{N-1} \bar{H}_m[k] = 1$. The mel-cepstrum computed with $H_m[k]$ or $\bar{H}_m[k]$ will differ by a constant vector for all inputs, so the choice becomes unimportant when used in a speaker recognition system that has been trained with the same filters. Figure 3.5 shows the filter bank frequency responses for the normalized $\bar{H}_m[k]$ and unnormalized $H_m[k]$ case.



(a) (b)

Figure 3.5: Mel filter bank frequency responses. (a) Normalized $\bar{H}_m[k]$ and (b) Un-normalized $H_m[k]$.

Let's define f_l and f_h to be the lowest and highest frequencies of the filterbank in Hz, F_s the sampling frequency in Hz, M the number of filters, and N the size of the FFT. The

boundary points $f[m]$ are uniformly spaced in the mel-scale, that is

$$f[m] = \left(\frac{N}{F_s}\right) B^{-1}\left(B(f_l) + m\frac{B(f_h) - B(f_l)}{M+1}\right) \tag{3.32}$$

where the mel-scale B is given by

$$B(f) = 2595\log(1 + f/700) \tag{3.33}$$

and B^{-1} is its inverse, given by the formula

$$B^{-1}(b) = 700(\exp(b/2595) - 1) \tag{3.34}$$

We then compute the log-energy at the output of each filter as

$$S[m] = \log\left(\sum_{k=0}^{N-1} |X[k]|^2 H_m[k]\right), \quad 0 < m \leq M \tag{3.35}$$

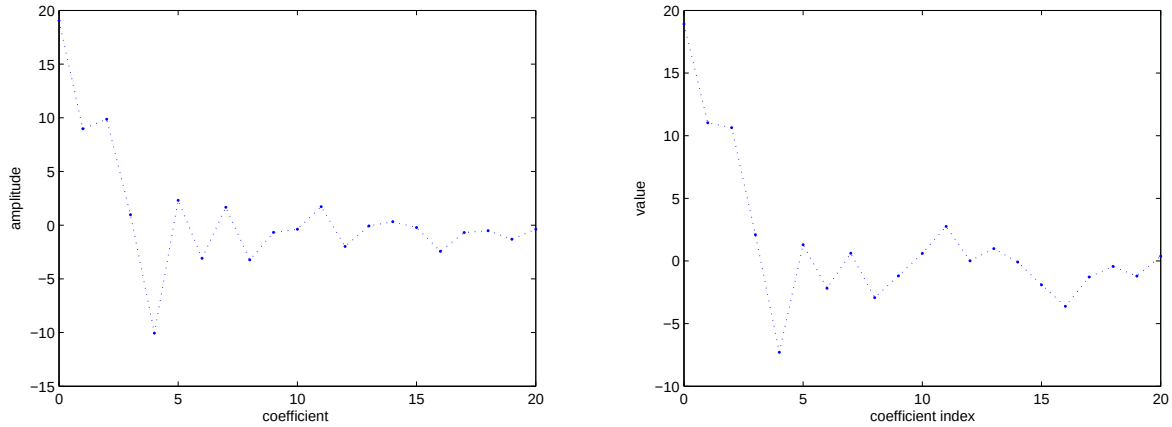
The mel-cepstrum is then the discrete cosine transform of the M filter outputs

$$c[n] = \sum_{m=0}^{M-1} S[m]\cos\left(\frac{\pi n(m - 1/2)}{M}\right), \quad 0 \leq n < M \tag{3.36}$$

where M varies for different implementations from 10 to 30. For speaker recognition, we have used the first 20 cepstrum coefficients. In Figure 3.6 we see the cepstral coefficients extracted with the aforementioned procedure for two different speech frames.

3.7.2 Sub-Cepstrum

Even though MFCC is extensively used for speaker recognition, subband based cepstral coefficients have attracted much attention over the past few years. The comparison between the two methods for extracting cepstral coefficients have shown that subband based methods have superior results for the speaker recognition task. The method differs slightly



(a) (b)

Figure 3.6: Cepstral coefficients extracted from two different speech frames.

form that described above. The only difference is that the filtering process is made in the time domain. That is, the speech signal $x[n]$ is passed through a bank of filters.

$$\tilde{x}[m,n] = x[n] * h_m[n] \tag{3.37}$$

where $m = 1 \ldots M$, and M is the number of filters. Each of the filters $h_m[n]$ is the inverse Fourier transform of each triangular frequency response $H_m[k]$, $(0 \leq k \leq N/2$, N is the size of the FFT).

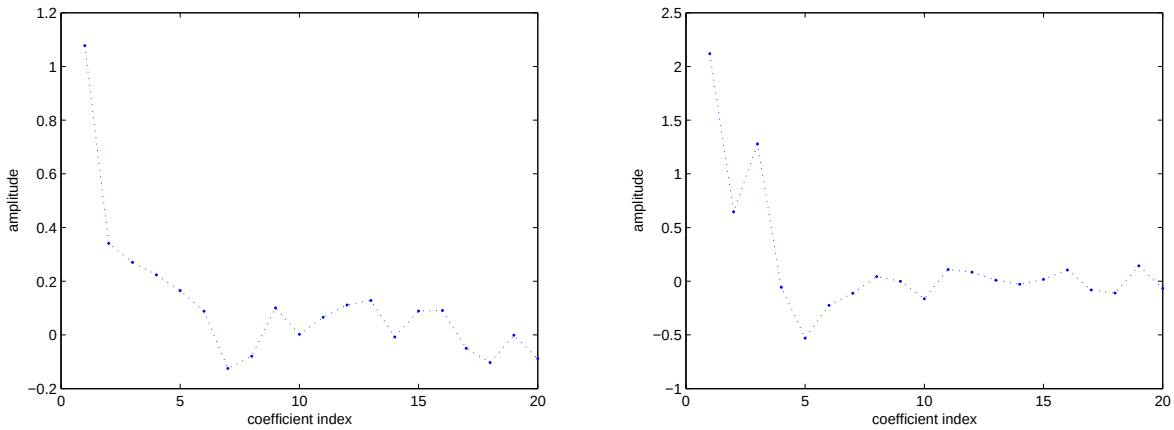
The output of the filtering process is windowed by a smoothing filter (e.g. a hamming window) and the energy of each frame is computed.

$$E[m,r] = \sum_{n=N/2}^{N/2} w[r-n]\tilde{x}[m,n] \tag{3.38}$$

Finally, the log energies for each filter and corresponding frames are cosine transformed to give the sub-cepstal coefficients.

$$c_{\text{sub}}[n] = \sum_{m=0}^{M-1} \log\left(E[m,r]\right) \cos\left(\frac{\pi n(m-1/2)}{M}\right) \tag{3.39}$$

In Figure 3.7 we see the sub-cepstral coefficients extracted with the aforementioned procedure for two different speech frames.



(a) (b)

Figure 3.7: Sub-cepstral coefficients extracted from two different speech frames.

Chapter 4

Experimental Results and Implementation Details

4.1 Introduction

This chapter presents the experimental results of the text-independent speaker identification and verification system. The goal of the experiments is to evaluate the performance of the system for pure (i.e. clean) speech and for speech degraded by different types of noise encountered in real life applications. Specifically, we evaluate the performance of a speaker identification system and a speaker verification system using three different types of features extracted from the speech signal. These features include, firstly cepstral coefficients using a nonlinear frequency axis following the Bark scale, secondly mel-cepstral coefficients warping the frequency axis using the mel-scale and finally, sub-cepstal coefficients which is a alternate method to compute spectral features. We also give information about the database of speakers used in the experiments.

4.2 Speech Database

The experiments for the speaker identification and verification system were conducted using a database of 47 individuals created in the eNTERFACE '05 workshop in Mons/Belgium [16]. The eNTERFACE database is a multilingual, multimodal database. Apart from speech data from different people around the world, it contains signature data and face data for each person. We are interested only in the speech data for our experiments. For each speaker there are six sessions of reading speech, that is each speaker is prompted with an appropriate text that can read loudly. Half of the sessions are in the native language (mother tongue) of the speaker and half is in the English language. Each session contains about 30 *sec* of speech. Therefore, we totally have 3 *min* of speech for each speaker. The speech from one session was recorded by a high quality microphone in a silence room. So, the speech recorded is of high quality (clean). For experiments we have used two sessions for training and the remaining for testing data. Namely, 1 *min* of speech is used for training data. The speech recorded has 16 *kHz* sampling frequency with 16 *bits* encoding accuracy.

4.3 Speaker Dependent Features

4.3.1 Extraction of Cepstral Coefficients

The input speech signal is segmented into frames of 20 *ms*, with 10 *ms* frame rate. That is, the second half of the current frame is the same as the first half of the next frame. A hamming window $w[n]$ of the same duration, 20 *ms*, is multiplied by the speech frame $s[n]$ to produce a short-time speech segment for analysis. The discrete Fourier spectrum of the

speech segment is obtained via the fast Fourier transform (FFT) from which the magnitude spectrum $|X(n, \omega_k)|$ is only kept, the phase is ignored. In mathematical terms the above procedure can be described by the following formula which is known as short time Fourier transform (STFT)

$$X(n, \omega_k) = \sum_{m=-\infty}^{\infty} x[m]w[n-m]e^{-j\omega_k m} \tag{4.1}$$

where $\omega_k = \frac{2\pi}{N}k$ with N the FFT length. The magnitude spectrum $|X(n, \omega_k)|$ is then put through a bank of triangular filters (see Figure 3.5). Let us denote each filter in the filterbank by $V_l(\omega_k)$, $l = 1, \ldots, L$. The center frequencies of the triangular bandpass filters follow the *mel-scale* or the *bark scale*. The number of filters L is selected to cover the entire signal bandwidth, that is from $0 - 8000$ *Hz*.

The energy output of each filter in the filterbank is then computed by

$$E(n, l) = \sum_{k=0}^{N-1} |X(n, \omega_k)|^2 \; V_l(\omega_k) \tag{4.2}$$

where N is the size of the FFT. We have assumed that the bandpass filters $V_l(\omega_k)$ are normalized according to their varying bandwidths, that is each filter satisfies $\sum_{k=0}^{N-1} V_l(\omega_k) = 1$.

The cepstral coefficients are are then the discrete cosine transform (DCT) of the L filter outputs, by firstly transform them in the log domain

$$C(n, m) = \frac{1}{L} \sum_{l=1}^{L} \log\{E(n, l)\} \cos\left(\pi m \frac{(l + 1/2)}{L}\right) \tag{4.3}$$

where $m = 0, \ldots, L-1$. The $C(n, 0)$ coefficient reflects the average energy in the speech frame n and is discarded as a form of amplitude normalization.

4.3.2 Extraction of Sub-cepstral Coefficients

In the previous section, we saw that STFT magnitude of every speech segment $x[n]$ was filtered by $V_l(\omega_k)$ as given in Equation (4.2). To be more accurate this is not a filtering operation on the signal $x[n]$. To understand this point, we note that the STFT can be viewed as a filtering operation where the analysis window $w[n]$ plays the role of the filter impulse response, that is

$$X(n, \omega_k) = e^{-j\omega_k n} \left(x[n] * w[n] e^{j\omega_k n} \right) \tag{4.4}$$

where $*$ denotes convolution. Specifically, $x[n]$ is passed through a bandpass filter whose impulse response is the analysis window $w[n]$ modulated to frequency ω_k. It follows from the filtering view of the STFT in Equation (4.4) that the speech signal $x[n]$ is first filtered by $w[n]e^{j\omega_k n}$ that has a narrow bandwidth. Therefore, the temporal resolution of the mel-scale filter energies computed in Equation (4.2) is limited by the STFT filters $w[n]e^{j\omega_k n}$.

An alternative method to compute the cepstral features is to convolve the mel-scale filter impulse responses directly with the speech signal $x[n]$. That is to make the filtering operation in the time domain. The output of each filter can be expressed by the following convolution operation

$$\tilde{X}_l(n) = x[n] * v_l[n] \tag{4.5}$$

where $v_l[n]$ denotes the impulse response corresponding to the filter frequency response $V_l(\omega_k)$ (see Figure (3.5)), of the l th mel-scale filter centered at frequency ω_l in the filterbank. The filter impulse responses are constructed in analytic signal form by requiring $V_l(\omega_k) = 0$ for $-\pi < \omega_k < 0$. That is, we firstly set to zero the negative frequencies of each bandpass filter $V_l(\omega_k)$ in the filterbank, and then we compute the analytic signal $v_l[n]$ via the inverse

Fourier transform. The filter impulse responses $v_l[n]$ are also referred to as *subband filters*. Examples of impulse responses $v_l[n]$ for low and high center frequency ω_l in the filter bank is depicted in Figure (4.1). The energy of the output of l th subband filter is computed by

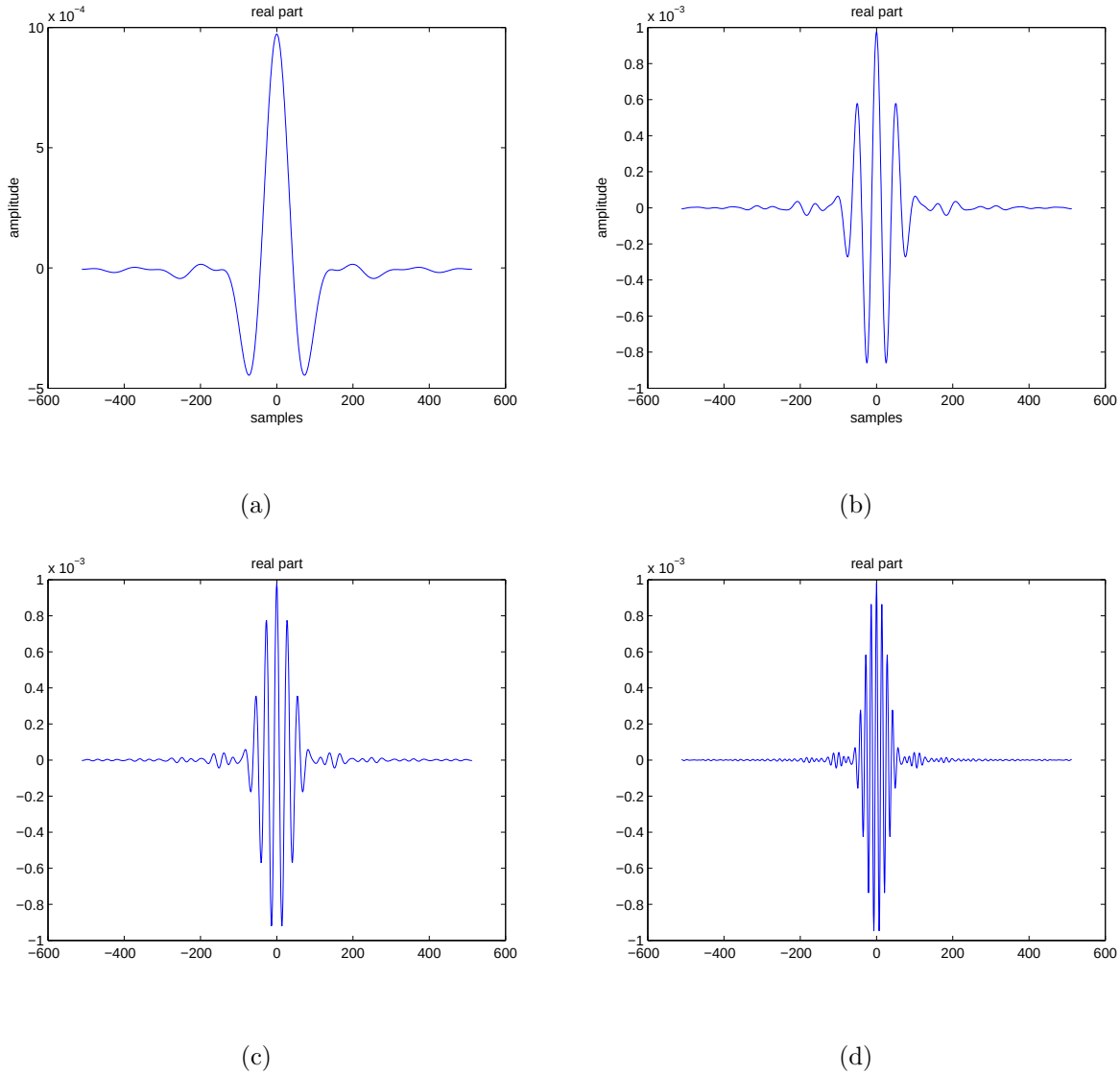


Figure 4.1: Impulse responses for the 2nd, 3rd, 10th and 13th filter in the filterbank.

$$E(n,l) = \sum_{m=-N/2}^{N/2} |X_l(m)|^2 \, p[n-m] \tag{4.6}$$

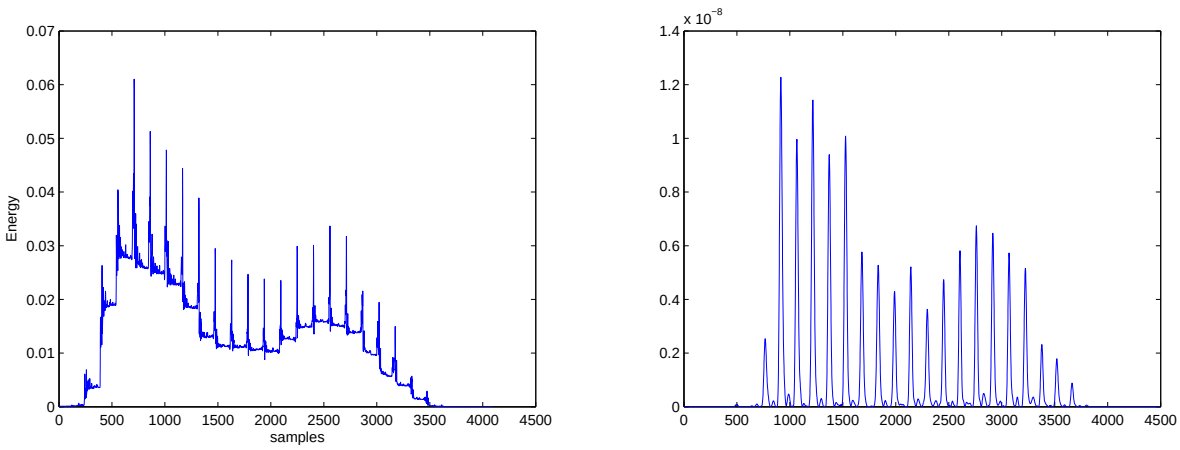
where $p[n]$ is a hamming window and N is the size of the window. The subband cepstral

coefficients are then computed by taking the DCT of the logarithm of the energies $E(n,l)$

$$C(n,m) = \frac{1}{L}\sum_{l=1}^{L} \log\{E(n,l)\} \cos\left(\pi m \frac{(l+1/2)}{L}\right) \tag{4.7}$$

where L is the number of filters in the filterbank.

The energy signals $E(n,l)$ can capture more temporal characteristics of the signal then the mel-scale filter energies, particularly for high frequencies. This is because, unlike in the mel-scale filtering operation, subband filters are applied directly to the signal $x[n]$. The difference in temporal resolution of the mel-scale filter and subband filter energies is illustrated in Figure 4.2



(a) (b)

Figure 4.2: Energies from mel-scale and subband filter banks, (a) mel-scale filter energy from filter number 14 ($\approx$ 3100 Hz), (b) subband filter energy from the same filter number.

4.4 Speaker Models

The next step in a speaker recognition system, whether for identification or verification, is to build a *model* of the voice of each speaker, using speaker-dependent features extracted from the speech waveform, as we have seen in the previous section. The basic idea is to use GMM to represent speakers. More specifically, the distribution of feature vectors extracted from a person's speech is modeled by a Gaussian mixture density. In particular, the mixture density for a speaker s is defined as

$$p(x|\lambda_s) = \sum_{i=1}^{M} a_i f_i(x) \tag{4.8}$$

where x is the feature vector variable, and $f_i(x)$ is a Gaussian distribution function. a_i are the weights that satisfy $\sum_{i=1}^{M} a_i = 1$ to ensure that $p(x|\lambda_s)$ is a legitimate density function. So, each speaker is parameterized by a set of weighted Gaussian functions in the feature space, namely $\lambda_s = \{a_i, \mu_i, \Sigma_i\}$, $i = 1, \ldots, M$, where μ_i and Σ_i is the mean values and the covariance matrices for the Gaussian functions $f_i(x)$. Speaker model parameters are estimated using the iterative Expectation-Maximization (EM) algorithm [35].

4.5 Identification System

The identification system is a simple maximum a posteriori classifier. We have a number of speakers represented by the models $\lambda_1, \ldots, \lambda_S$, and we want to identify a speaker given as input their voice. The objective is to find the speaker model which has the maximum posterior probability for the input feature vector sequence, $x_1, \ldots, x_T$, where x_i is a feature vector extracted from a speech frame. Using the Bayes' formula, we choose the speaker

such that

$$\begin{aligned}\hat{s} &= \arg\max_{1\le s\le S} P(\lambda_s|x_1,\ldots,x_T) \\ &= \arg\max_{1\le s\le S} \frac{p(x_1,\ldots,x_T|\lambda_s)P(\lambda_s)}{p(x_1,\ldots,x_T)}\end{aligned} \tag{4.9}$$

Assuming that each speaker is equally likely to appear, that is we are assuming equal prior probabilities $P(\lambda_s)$. The term $p(x_1,\ldots,x_T)$ is constant for all speakers. Furthermore, assuming independence among feature vectors, the last formula becomes

$$\hat{s} = \arg\max_{1\le s\le S} \prod_{t=1}^{T} p(x_t|\lambda_s) \tag{4.10}$$

and using logarithms we have

$$\hat{s} = \arg\max_{1\le s\le S} \sum_{t=1}^{T} \log p(x_t|\lambda_s) \tag{4.11}$$

where $p(x_t|\lambda_s)$ is given by Equation (4.8). Table 4.1 shows the results of the identification system for a number of tests.

Table 4.1: Results for the identification system, the test speech is about 2 *sec.*

Test #	**Mel-cepstrum**	**Sub-cepstrum**
1	95.7%	100%
2	93.6%	95.7%
3	91.4%	95.7%
4	95.7%	97.8%

4.6 Verification System

Given an input voice and a hypothesized speaker, S, the task of speaker verification, is to determine if the input speech was spoken by S. That is, the system must decide if the input voice came from the hypothesized speaker or not. For a given input voice and a hypothesized the choice is between H_0 and H_1

H_1**:** The input voice is from the hypothesized speaker

H_0**:** The input voice is **not** from the hypothesized speaker

In order to perform the likelihood ratio test we need some model for the universe of all possible impostor speakers. For this purpose we pool speech from all speakers and train a single model, and we call it the universal background model (UBM).

The approach used in the speaker verification system is to apply a likelihood ratio test to the input voice to determine if the input speaker is accepted or rejected. For the input feature vector sequence $x_1, \ldots, x_T$, extracted from the input speech, and a claimed speaker identity with corresponding model λ_C, the likelihood ratio is

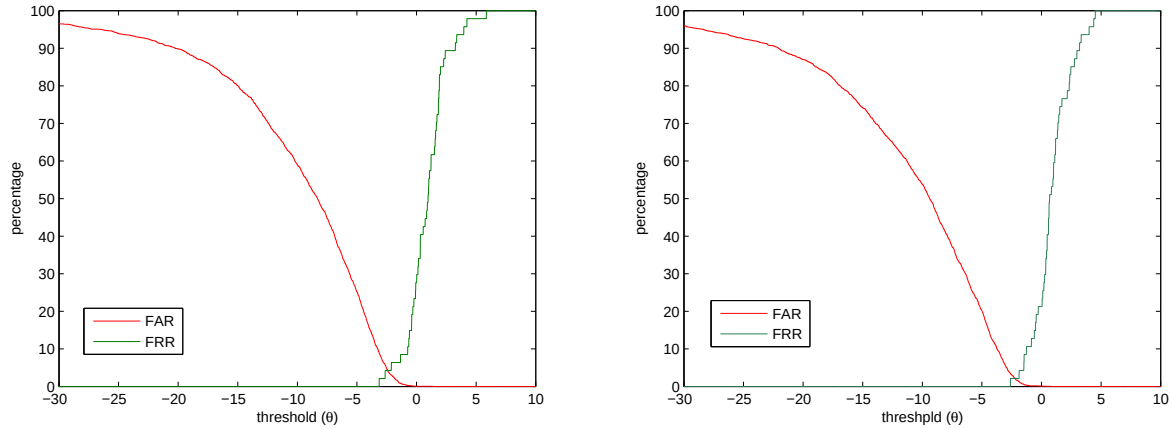
$$\Lambda(x_1, \ldots, x_T) = \frac{P(\lambda_C | x_1, \ldots, x_T)}{P(\lambda_{\text{UBM}} | x_1, \ldots, x_T)} \tag{4.12}$$

Applying Bayes' rule as in the previous section, the likelihood ratio in the log domain becomes

$$\begin{aligned} \log \Lambda(x_1, \ldots, x_T) &= \log p(x_1, \ldots, x_T | \lambda_C) - \log p(x_1, \ldots, x_T | \lambda_{\text{UBM}}) \\ &= \sum_{t=1}^{T} \log p(x_t | \lambda_C) - \sum_{t=1}^{T} \log p(x_t | \lambda_{\text{UBM}}) \end{aligned} \tag{4.13}$$

The above likelihood ratio is compared with a threshold θ and the claimed speaker is accepted if $\Lambda(x_1, \ldots, x_T) > \theta$ or rejected if $\Lambda(x_1, \ldots, x_T) \leq \theta$. The threshold θ is adjusted

such that there is a trade-off between rejecting genuine claimants (false rejection) and accepting impostors (false acceptance). Figure 4.3 shows how we can find the optimal threshold θ. We choose the threshold where the false acceptance rate (FAR) and the false rejection rate (FRR) curve intersect.



(a) (b)

Figure 4.3: (a) Verification using mel-cepstrum, (b) Verification using sub-cepstrum.

4.7 Identification Under Adverse Conditions

In many practical applications a speaker identification system must operate under adverse conditions. This means that the speech used to test an identification system may have been corrupted by environmental noise. As discussed above, the Gaussian mixture model is based on modeling the underlying distribution of feature vectors from a speaker. When the speech is corrupted, the spectral based features are also corrupted and so their distributions are modified. Thus, a speaker model trained using clean speech from a controllable environment will greatly performed poorly in recognizing the same speaker using speech collected under

different conditions, as it is often the case, since the feature distributions are now different. Experiments of speaker recognition using degraded speech have shown a dramatic decrease in performance. To maintain speaker recognition rates at an acceptable level, some form of compensation must be applied. We have used Karhunen-Loeve transform (KLT) to remove the degradation effect, but the results are not encouraging to adopt such an technique for real life applications. The experiments are primarily conducted with corruptions due to additive acoustic noise from various sources. Table 4.2 shows the identification results for different sorts of noise. The experiments were conducted with 5 *sec* of testing speech and SNR = 10 dB.

Table 4.2: Results for the identification system under adverse conditions.

Noise type	**Performance**
babble	91.4%
engine	38.2%
cockpit	29.7%
factory	59.5%
machine gun	95.7%
military vehicle	100%
pink	78.7%
tank	97.8%
car interior	87.2%
white	53.1%

Chapter 5

Concluding Remarks

5.1 Conclusion

It is useful to examine the lack of commercial applications until very recently for automatic speaker identification and verification. Speaker identification and verification analyze speech signals to extract spectral parameters such as cepstral coefficients. Speaker identification has the objective of selecting which of N speakers spoke. However, our understanding of how listeners exploit spectral cues to identify human sounds far exceeds our current knowledge of how to distinguish speakers. For speaker identification, using GMM methods yield good results in limited tests, but performance decreases under adverse conditions that might be found in practical applications. For example telephone line distortions, uncooperative speakers, and speaker variability over time often lead to accuracy levels unacceptable for many applications.

Modern speaker identification applications now use nonlinear techniques. The more difficult task of speaker identification is often impractical because of the tendency of increasing error probability as the population size increases. Even for large populations, current speaker identification techniques appear to yield sufficient accuracy for practical applications where high-quality speech is available or when other biometric are also used.

5.2 Future Work

In this thesis we have described the major elements of the GMM system used for high-accuracy speaker recognition. The GMM system is built around the optimal maximum likelihood test for identification, using simple but effective Gaussian mixture models for likelihood functions.

While the GMM system has proved to be very effective for speaker recognition tasks, there are several open areas where future research can improve or build on from the current approach. The first area is dealing better with speaker specific features. The GMM system, and all current speaker state-of-the-art recognition systems, rely on low-level acoustic information. Unfortunately, speaker and channel information are bound together in an unknown way in the current spectral-based features and the performance of these systems degrades when the microphone or acoustic environment changes between training data and testing data. Progress must be made in minimizing this weakness.

The second area is incorporating higher levels of information, such as speaking style supra-segmental features, or word usage, into the decision making process. Humans use several levels of information to recognize speakers from speech alone, but automatic systems are still dependent on the low-level acoustic information. The challenges in this area are to find, reliably extract, and effectively use these higher levels of information from the speech signal. It is likely that these higher levels of information will not provide good performance on their own and may need to be fused with more traditional acoustic-based information.

References

[1] Douglas A. Reynolds, Thomas F. Quatieri, and Robert B. Dunn, "Speaker Verification Using Adapted Gaussian Mixture Models," *Digital Signal Processing*, vol. 10, pp. 1941, 2000.

[2] G. Doddington, "Speaker recognition: Identifying people by their voices," *Proceedings IEEE*, vol. 73, pp. 1651-1664, 1985.

[3] R. Mammone, X. Zhang, and R. Ramachandran, "Robust Speaker Recognition," *IEEE Signal Processing Magazine*, vol. 13, no. 5, pp. 58-71.

[4] Douglas A. Reynolds, and Richard C. Rose, "Robust Text-Independent Speaker Identification Using Gaussian Mixture Speaker Models," *IEEE Transactions on Speech and Audio Processing*, vol. 3, no. 1, January 1995.

[5] Richard O. Duda, Peter E. Hart, David G. Stork, *Pattern Classification*, John Wiley & Sons, Second Edition, 2001.

[6] Gin-Der Wu and Chin-Teng Lin, *Word Boundary Detection with Mel-scale Frequency Bank in Noisy Environment*, IEEE Transactions on Speech and Audio Processing, Vol. 8, No. 5, September 2000.

[7] Sadaoki Furui, *Recent advances in speaker recognition*, Pattern Recognition Letters, 18, 859-872, 1997.

[8] Thomas F. Quatieri, *Discrete-Time Speech Signal Processing, Principles and Practice*, Prentice Hall PTR, Massachusetts Institute of Technology, Lincoln Laboratory, 2002.

[9] Douglas A. Reynolds, *Speaker identification and verification using Gaussian mixture speaker models*, Speech Communication, 17, 91-108, 1995.

[10] Douglas A. Reynolds, *A Gaussian Mixture Modeling Approach to Text-Independent Speaker Identification*, Georgia Institute of Technology, Ph.D. Thesis, August 1992.

[11] Sergios Theodoridis and Konstantinos Koutroumbas, *Pattern Recognition*, Elsevier Academic Press, Second Edition, 2003.

[12] Douglas O'Shaughnessy, *Speech Communications, Human and Machine*, IEEE Press, Second Edition, 2000.

[13] A. Higgins, L. Bahler, and J. Porter, *Speaker verification using randomized phrase prompting*, Digital Signal Processing, 1, 89-106, 1991.

[14] C. Veilhauer, S. Schimke, A. Valsamakis, and Y. Stylianou, *Fusion Strategies for Speech and Handwriting Modalities in HCI*, Proceedings of SPIE-IS&T Electronic Imaging, Vol. 5684, pp.63-71, ISBN 0-8194-5657-8, 2005.

[15] S. Schimke, A. Valsamakis, C. Vielhauer, Y. Stylianou, *Biometrics: Different Approaches for Using Gaussian Mixture Models in Handwriting*, Proceedings of Communications and Multimedia Security (CMS 2005), pp. 261-263, Lecture Notes in Computer Science, ISBN 3-540-28791-4, 2005.

[16] Y. Stylianou, Y. Pantazis, F. Calderero, P. Larroy, F. Severin, S. Schimke, R. Bonal,

F. Matta, and A. Valsamakis, *GMM-Based Multimodal Biometric Verification*, Final Report for the Proceedings of eNTERFACE'05 (Similar NoE Workshop on Multimodal Interfaces, Mons/Belgium, July 18th - August 12th, pp. 44-51, ISBN 2-87463-003-9, 2005.

[17] Steven B. Davis, and Paul Mermelstein, "Comparison of Parametric Representations for Monosyllabic Word Recognition in Continuously Spoken Sentences," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, Vol. ASSP-28, No. 4, pp. 357-366, August 1980.

[18] J. Godfrey, D. Graft, A. Martin, "Public databases for speaker recognition and verification," *ESCA Workshop on Automatic Speaker Recognition, Identification and Verification*, pp. 39-42, 1994.

[19] A. Rosenberg, "Automatic speaker verification," *Proceedings IEEE*, vol. 64, pp. 475-487, 1976.

[20] B. Atal, "Automatic recognition of speakers from their voices," *Proceedings IEEE*, vol. 64, pp. 460-475, 1976.

[21] K. Wu, and D. Childers, "Segmentation of speech using speaker identification," *Proceedings IEEE Int. Conf. ASSP*, vol. 1, pp. 161-164, 1991.

[22] K. Yu, J. Mason, and J. Oglesby, "Speaker Recognition Models," *Proc. Europ. Conf. Speech Comm. & Tech.*, pp. 629-632, 1995.

[23] H. Matsui, and S. Furui, "Comparison of text-independent speaker recognition meth-

ods using VQ-distortion and discrete/continuous HMM's ," *IEEE Transactions Speech & Audio Processing*, vol. 2, pp. 456-459, 1994.

[24] H. Gish, and M. Schmidt, "Text-independent speaker identification," *IEEE Signal Processing Magazine*, vol. 11, no. 4, pp. 18-32, 1994.

[25] L. Liu, J. He, and G. Palm, "A comparison of human and machine in speaker recognition," *Proc. Europ. Conf. Speech Comm. & Tech.*, pp. 2327-2330, 1997.

[26] B. Atal, "Effectiveness of linear prediction characteristics of the speech wave for automatic speaker identification and verification," *J. Acoust. Soc. Am.*, vol. 55, pp. 1304-1312, 1974.

[27] F. Soong, and A. Rosenberg, "On the use of instantaneous and transitional spectral information in speaker recognition," *IEEE Trans. ASSP*, vol. ASSP-36, pp. 871-879, 1988.

[28] J. Openshaw, Z. Sun, and J. Mason, "A comparison of composite features under degraded speech in speaker recognition," *Proc. IEEE Int. Conf. ASSP*, vol. II, pp. 371-374, 1993.

[29] A. Rosenberg, and F. Soong, "Recent research in automatic speaker recognition," *Advances in Speech Signal Processing*, pp. 701-738, 1992.

[30] J. Fakotakis, and J. Sirigos, "A high performance text independent speaker recognition system based on vowel spotting and neural nets," *Proc. IEEE Int. Conf. ASSP*, pp. 661-664, 1996.

[31] B. Necioglu, M. Clements, and T. Barnwell, "Objectively measured descriptors applied to speaker characterization," *Proc. IEEE Int. Conf. ASSP*, pp. 483-486, 1996.

[32] "http://www.clsp.jhu.edu/ws2002/groups/supersid," *SuperSID Project at the JHU Summer Workshop*, July-August 2002.

[33] A. Rosenberg, and S. Parthasarathy, "Speaker background models for connected digit password speaker verification," *Proc. IEEE Int. Conf. ASSP*, pp. 81-84, 1996.

[34] D. Reynolds, "Comparison of background normalization methods for text-independent speaker verification," *Proc. Europ. Conf. Speech Comm. & Tech.*, pp. 963-966, 1997.

[35] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *Journal of the Royal Statistical Society, Series B (Methodological)*, vol. 39, no. 1, pp. 1-38, 1977.

[36] R. Redner, and H. Walker, "Mixture densities, maximum likelihood and the EM algorithm," *SIAM Review*, vol. 26, no. 2, 1984.

[37] M. Jordan, and R. Jacobs, "Hierarchical mixtures of experts and the EM algorithm," *Neural Computation*, vol. 6, pp. 181-214, 1994.