UNIVERSITY OF CALGARY

A Novel Psychoacoustic Approach to Speaker Recognition

by

Lani Dee Letty Bateman

A THESIS

SUBMITTED TO THE FACULTY OF GRADUATE STUDIES

IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE

DEGREE OF MASTER OF SCIENCE

DEPARTMENT OF COMPUTER SCIENCE

CALGARY, ALBERTA

APRIL, 2005

UNIVERSITY OF CALGARY

FACULTY OF GRADUATE STUDIES

The undersigned certify that they have read, and recommend to the Faculty of Graduate Studies for acceptance, a thesis entitled "A Novel Psychoacoustic Approach to Speaker Recognition" submitted by Lani Bateman in partial fulfilment of the requirements of the degree of Master of Science.

Supervisor, Jim Parker, Computer Science

Jörg Denzinger, Computer Science

Gail Kopp, Education

Apr. 26, 2005
Date

Abstract

Linear Prediction (LP) is a low-dimensional method of representing speech that is commonly used for both speech and speaker recognition. There are many alternative representations of LP that outperform it in recognition tasks, one of which is referred to as Line Spectrum Pair (LSP) frequencies. An extension of LP that more closely models what is happening in the human ear is a method called Perceptual Linear Prediction (PLP). This method has also been shown to outperform the LP method. In this thesis I will show that it is possible to represent the PLP parameters in the alternative LSP representation that is analogous to that of the LP method, creating a more powerful set of psychoacoustic parameters for recognition purposes, for both speaker and speech recognition applications.

Table of Contents

List of Tables

List of Figures and Illustrations

List of Symbols, Abbreviations and Nomenclature

ANN	Artificial Neural Network
ASR	Automatic Speech Recognition
df	Degrees of Freedom
DTW	Dynamic Time Warping
FAR	False Accept Rate
FRR	False Reject Rate
GMM	Gaussian Mixture Model
HMM	Hidden Markov Model
LPC	Linear Prediction Coefficients
LSP	Line Spectrum Pairs
PDF	Probability Density Function
pdf	Probability Distribution Function
PLP	Perceptual Linear Prediction
PLP-LSP	Perceptual Linear Predictive Line Spectrum Pairs
VQ	Vector Quantization

CHAPTER ONE: BIOMETRICS

1.1 Introduction

The human body is an amazing and unique thing. It can truly be said that the mold is broken after each one of us is born; not two of us are exactly alike. There are several distinguishing characteristics that each one of us possesses that make us different from everybody else. The human body houses various "signatures" that can be used to distinguish one person from another. What is even more amazing is that the human brain has the ability to identify, process and analyze, and make use of much of this information.

Pattern recognition is the process of extracting information out of the environment around us for some purpose. Human beings make most of their critical decisions based on patterns. Meaningful information is collected and analyzed from the input our senses provide. When a sound is made and a person hears it, the following happens: the "sound" comes in the form of pressure waves. These waves are carried through the air and make contact with the ear. They travel through the outer, middle, and inner ears, where they encounter auditory structures that convert the sound wave energy to electrical energy, and the information that is extracted is passed up to the brain through a series of complex neural pathways.

It is desirable to be able to process and analyze vast amounts of information using a computer in the same way that a human being is capable, and to the same degree of accuracy, or better. But the process of pattern matching can be extremely complex,

particularly in the case of speech and speaker recognition, and not enough is understood about how it should be done. Since the human body is an already working model that is able to perform many of the tasks that we would like to automate, it makes a great deal of sense to simulate these human pattern recognition and decision-making abilities using what knowledge we have, and adding to this as our knowledge of how the human body works increases. Unfortunately, there is still so much that is unknown about the inner workings of the brain and the complex neural pathways that carry information to higher brain centers. The processes that occur from the ear to the brain and within the brain itself are still largely a mystery. This lack of information about how the human body works prevents us from adequately modeling these processes. As advances in biology and psychology are made, however slowly, these mysteries are being solved little by little, but there is still a vast amount that needs to be discovered. Luckily, engineers, statisticians, and mathematicians are not without their tricks. Signal processing, probability and statistics, and data mining techniques are some of the powerful tools that are used by researchers for pattern matching.

1.2 Biometrics

Biometrics are measurable biological data that can be reliably reproduced by the person from whence it came. In particular, these biometrics are those target "signatures" of an individual that make them unique. Biometrics are typically analyzed statistically to distinguish one person from another, which means that the biological data must be obtained such that it uniquely characterizes that person. This data can be a measurement of a physical feature or repeatable behavior of the individual. Pattern matching can be

performed on biometrics in order to decide which biometrics are unique to a given individual.

Physical biometrics are measurements of physical features of a person. These features cannot be changed through a conscious mental effort on the part of the individual, although certain behaviors can, in fact, change the physical characteristics over time, such as smoking, poor nutrition, or poor posture. Because physical biometrics are reliably reproducible, they are somewhat more reliable than behavioural biometrics, which are discussed next. Some physical biometrics that are commonly used for recognition purposes include the iris, fingerprints, face, and hand and finger geometry.

The behavioural characteristics of a person are the repeatable aspects of their actions and behaviors that can be measured and analyzed for unique identification. Speaking, hand-writing, and walking are examples of such actions. These measurable behavioural characteristics of a person are what define the behavioural aspect of biometrics. Achieving high recognition and verification of an individual based on behavioral biometrics is somewhat more difficult than using physical biometrics. Behaviors can be changed dramatically without much effort on the part of the individual, and can even be changed involuntarily and without the individual's knowledge. Even if someone tries very hard to reproduce the same behavior, it will never manifest itself in exactly the same way twice. Therefore there is a great deal more variation in the behavioral biometrics of a person, making it harder to capture and recognize these behaviors. In addition, behaviors will gradually change over time as a person continually adapts to changing environments

and preferences, as well as physical conditions such as the effects of aging. Temporary behavioral changes may come about due to illness or a change in mood. The behavioral biometric of speech is the focus of this thesis.

1.3 Speech Analysis

Speech is a complex grouping of sounds, and contains a lot of various pieces of information. It contains both information about what is being said and who is saying it. The information that a person is trying to communicate is the primary characteristic of speech. The meaning conveyed within the speech is referred to as lexical content, and includes the words and vocabulary of the speech as well as the language that is being spoken. The way in which something is said is also apparent. It is sometimes possible to tell whether a speaker is happy, sad, or angry, and whether a person is tired, healthy, or sick. In addition to what a person is saying, speech also contains information about who is speaking. The characteristics of someone's voice are unique to that person. When a friend calls you on the phone, you recognize who they are even though you can't see them: you recognize them by their voice. Other information about the person talking is also evident, such as whether the person is male or female, whether they are young or old, whether they smoke or not, and what kind of accent they have.

The goal of speech analysis is to extract one or more of the aforementioned pieces of information for some useful purpose. For example, it might be useful for someone to be able to dictate a letter to a computer by speaking into a microphone and have the computer translate the spoken words into text and place them into a word processor

document. Similarly, as a security measure, it might be useful to have a secure system verify the identity of a person by their voice before giving them access to sensitive data.

As the first step of speech analysis, an utterance is produced by a speaker and is captured in the form of a waveform. An utterance is a complete unit of talk, bounded by silence. A waveform is a digital representation of the utterance that can be analyzed by a machine. Then speech signal processing is used to extract the aforementioned pieces of information from the waveform, which are referred to as "features" of the speech. These techniques are discussed in chapter 3. Using these features in some meaningful way is done using pattern matching and classification techniques, which are introduced in a general way in chapter 2 and are described in more detail in a speaker recognition context in chapter 4.

1.4 Problems Associated with Speaker Recognition

Because speech is a behavioral biometric, it is extremely variable and thus is hard to adequately characterize using computers. As already mentioned, there is still so much we don't know about speech production, and there is even more we don't know about how speech is processed by the human brain. Due to this lack of knowledge, and sometimes due to a lack of the necessary vast amount of computing power, current speaker recognition systems are far from perfect. The variability that is encountered naturally in speech can be a problem. The range of this variability can be somewhat broad and is difficult to capture and take into account. In addition, extraneous background noises are always mixed in with speech and it is extremely difficult, if not impossible, to isolate and

separate only the speech for analysis. These issues make it necessary to continue to find better ways of making speaker recognition work.

1.5 Overview of Thesis

In speech analysis, computations are performed to extract parameters, or features, that represent the speech and capture the characteristics of the speech that we are trying to represent, while ignoring those aspects of the speech that are unimportant. Four sets of parameters in particular are the focus of this thesis, all of which are described in detail in chapter 4. The first set of features are very commonly-used and are typically used in speech analysis. They are known as Linear Prediction Coefficients (LPC) (Makhoul, 1996), (Makhoul, 1975). They are obtained through a technique that is known as signal processing, which is discussed in chapter 3, and are modeled after a series of piecewise-joined acoustic tubes that crudely represent the vocal tract.

Another important set of parameters are known as Line Spectrum Pair (LSP) frequencies (Itakura, 1975), which are features that have originally been used for speech data compression. These features are an alternative representation of LPC parameters, meaning that they are computed directly from LPC parameters. As will be discussed in chapter 4, this alternative representation has many advantages over the LPC representation (Soong and Juang, 1984). As with all features, LSP frequencies are a low-dimensional representation of high-dimensional speech data. This low-dimensionality allows for quicker transfer of data across networks and through cellular phone technology (Campbell et al., 1991). To transfer the speech across a channel, the speech is first

compressed into LSP frequencies at one end. The LSP frequencies are then transferred to the other end, where they are reconstructed (or decompressed) into an approximation of the original speech.

It has been only in more recent years that these Line Spectrum Pair frequencies have been used for speaker recognition tasks. Because they are so highly effective in speech compression, they are very effective at characterizing the underlying speech and are therefore a good candidate for speaker recognition. Clearly, LSP frequencies have a lot of potential and their properties need to be further explored.

The third set of parameters come from psychoacoustics, which is a field of study in which the perceptual experiences of humans is combined with the observed physical behavior of the ear to create synthetic models of the human auditory system. While these models are somewhat limited, they more accurately represent the physical processes with the human ear. The parameters known as Perceptual Linear Prediction (PLP) coefficients (Hermansky, 1990) are based upon this research, and have a great deal in common with LP coefficients. PLP coefficients are simply an extension of the LP coefficients. The computation of the PLP coefficients requires several steps, the last of which is a similar procedure used to obtain the LPC parameters, as will be described in chapter 4. They have been shown to be more effective in speaker recognition tasks than LPC under certain conditions. The higher the number of PLP parameters used, the more speaker-specific voice characteristics are represented.

This thesis presents a novel representation of the PLP method, which is evaluated in a text-independent speaker identification task. It is the purpose of this thesis to see if the enhanced properties of the LSP frequencies can be successfully applied to the PLP method in the same way as in the LP method, thereby deriving a new, more powerful and robust psychoacoustic speaker recognition method. Because LP and PLP are used in both speaker and speech recognition, the benefits gained by the new PLP-LSP method would also carry over into the domain of Speech Recognition. In addition, since many of the other psychoacoustic methods share the same computational similarity with the LP method that the PLP method has, it is likely that this novel representation can be applied to those methods as well.

This thesis is organized as follows. Chapter 2 introduces the concepts of pattern-matching, data mining, and modeling and classification. Then the concepts of speech and speaker recognition are discussed.

Chapter 3 presents the mathematical theory and signal processing procedures that are used to analyze speech and extract meaningful information from it for recognition purposes.

In chapter 4, some of the most common recognition methods are surveyed, and the well-known LPC, LSP, and PLP features are discussed in more detail.

Chapter 5 presents an experiment in which the performance of the LPC, LSP, PLP, and extended PLP-LSP features are evaluated and compared in a text-independent speaker recognition task.

In chapter 6, the results of this experiment are discussed.

CHAPTER TWO: PATTERN MATCHING IN SPEECH

2.1 Pattern Matching and Classification

Researchers in electrical engineering and mathematics disciplines have done a good deal of research on the analysis of signals. A signal is a detectable physical quantity or impulse by which information can be transmitted. Speech can be thought of as a signal, which means that all the techniques used for signal processing can be used for analyzing speech. The patterns extracted from the speech can then be used with well-known pattern matching and classification techniques. Speech can also be thought of as "random", which means that probabilistic and statistical techniques can be applied to it (Stark and Woods, 2002). In fact, many signal processing and statistical methods are combined when performing pattern recognition on speech.

In pattern matching, the idea is to build a "model" of what we are trying to represent. A model is an object that can be made comparable to other objects to determine their similarity. Similarity measures usually come in the form of some sort of distance metric. We say that objects that are "close" to one another are more similar and objects that are further apart are less similar. To build a model, it is necessary to define a set of characteristics that the model should represent, so that objects of similar characteristics will be "closer" to one another than objects that have differing characteristics.

The characteristics, or attributes, of an object are usually extracted in the form of an ordered set of measurements. These values are referred to as features, and the group of

features is called a feature vector. A very simple example of a comparison that can be defined between these feature vectors is Euclidean distance.

For example, suppose we have three two-dimensional points in space (x, y): $c = (2,3)$, $d = (10,5)$, and $e = (1,1)$. The points can be thought of as feature vectors, which represent their physical position in Euclidean space. The Euclidean distance between two points a and b is defined as:

$$d(a,b) = \sqrt{(a_x - b_x)^2 + (a_y - b_y)^2} \tag{1.1}$$

Using this metric, the Euclidean distance between c and d is 8.24 units, the distance between c and e is 2.2361 units, and the distance between d and e is 9.8489 units. Here the units are not important. What matters are the relative distances between the different points. It is easy to see that points c and e are closer to each other than either one is to point d.

2.2 Classification

Classification is an important part of pattern matching. In classification, each observation, or member instance, within a data set is said to belong to one of a set of classes. A set of "predictor" attributes, which will be referred to as "features" in this thesis, and a "goal" attribute is assigned to each observation. The goal attribute indicates what class the

observation belongs to, and the features are the values that are used to predict the goal attribute.

The main idea is to define a set of rules that will characterize the relationship between the features and the goal attribute, using a set of known observations and their known goal attributes, in order to use this relationship to classify future observations that we have not yet seen, and whose goal attributes are unknown. These unknown goal attributes are predicted from their corresponding known features using the relationship derived between the known features and their corresponding (known) goal attributes that were used to define the relationship in the first place. The relationship we are trying to obtain obviously depends on how well suited the features are to the task at hand. For example, if the goal is to determine the diagnosis of someone's illness, the decision must be based on things like the person's symptoms and family medical history, not their name or social security number.

The process of defining a relationship between the features and the goal attributes of the observations is known as training, and the accuracy of the found relationship to actually predict the category, or class, of future observations is known as testing. A set of data is obtained and split randomly into two sets: the training set and the testing set. The set of data in the training set is used to discover the relationship between the features and their goal attributes, and the relationship that is discovered is then used to predict the goal attributes of the set of data in the testing set. The predicted goal attributes are compared to the actual goal attributes of the test set, which are effectively unknown instances that

the training phase has not seen, and based on how many goal attributes were correctly predicted, an accuracy is computed. This reveals how well the relationship that was selected actually determines goal attributes based on features. It is the goal of classification to maximize this accuracy in the test set, so that when real-life data is input to the system, where the goal attribute is not known in advance, it will be correctly classified.

2.3 Speech and Speaker Recognition

As already mentioned, speech contains information about both what is being said, and who is speaking. This is true because it is possible for a human being to determine each. A person is capable of knowing what is being said by someone whose voice they have never heard before, and are also capable of understanding what is being said by someone whose voice is familiar who has never said that phrase to them before.

Speech recognition and speaker recognition are two of the main goals of speech analysis. In speech recognition, the objective is to determine *what is being said.* It is usually desirable to use information about what is being said and to ignore information about who is speaking. In speaker recognition, the objective is to determine *who is speaking.* It is primarily desirable to use information about who is speaking and ignore information about what is being said. In some cases, however, it is also important to determine what is being said as a secondary goal.

2.3.1 Identification and Verification

In both speech and speaker recognition, there are two different objectives, which are identification and verification. In what follows, identification and verification will be discussed largely in the context of speaker recognition.

2.3.1.1 Speaker Recognition: Identification and Verification

Speaker recognition encompasses both speaker identification and verification. The goal of speaker identification is to identify an unknown speaker out of a group of possible known speakers. The known speakers have voice models which are stored in a database. These voice models represent the unique characteristics of each person's voice who has enroled in the system. Typically, an utterance belonging to the unknown person is presented to the speaker identification system. The utterance is analyzed and the resulting voice data is compared to the speaker models that are stored in the database of known speakers through a one-to-many matching process. The desired outcome is to find the speaker model that is most likely to have generated the unknown speaker's input utterance, thereby identifying the identity of the unknown speaker. It is also possible that none of the voice models matches the unknown speaker's voice. As a database increases in size, not only does the task become much more time-consuming and tedious, but the chances of finding more than one stored speaker whose voice model is very close to the unknown speaker also increases, creating ambiguity and increasing the rate of false identification.

A target unknown speaker may be unaware of his or her involvement in the speaker identification process. Depending on the situation, it may be assumed that the input speech is of variable quality. For example, the sample voice data could have been obtained over a telephone wire or in the middle of a noisy airport. Furthermore, multiple voices may have been captured in the same data, thereby making it harder to separate out the target individual's voice patterns. When a speaker is aware of his or her involvement, the quality of the speech tends to be higher since the identification process is more likely to occur in a more controlled environment. Speakers may consciously or unconsciously make an extra effort to enunciate more clearly and speak at a more audible volume.

The speaker verification problem is somewhat less computationally expensive than the speaker identification problem. Here, a speaker claims an identity and then a match is performed against the corresponding speaker model of the target speaker. In other words, a one-to-one matching process is performed, at which point a binary decision is made. If the distance, or distortion, between the utterance of the unknown speaker and the target voice model falls within a certain threshold of closeness to the target voice model, the speaker is accepted as having that identity, otherwise they are rejected. This type of system is used when someone wishes to gain access to some restricted service, such as logging on to a network or taking money out of a bank machine.

2.3.1.2 Open vs Closed Set Systems

In an open-set system, an unknown speaker may or may not be one of a set of known speakers, which are those with speaker models that are recognized by the system. This is

generally the case in forensic and police applications. Therefore it may be the case that a reference model for an unknown speaker may not exist in the database. In this situation it is necessary to provide a way for the system to decide that an unknown speaker is not close enough to any of the speaker models in the system, and be rejected outright. This is done by providing a threshold test that can be used to determine how close a match must be before the speaker is classified as being one of the known speakers. If the threshold test fails, the system may report that the speaker's identity could not be determined, that more speech input data is necessary for an adequate decision to be made, or that the speaker is definitely not one of the speakers recognized by the system. In a closed-set system, it is assumed that an unknown speaker is one of a set of speakers recognized by the system, meaning that all users of the system must have a speaker model in the database.

2.3.1.3 Text-Dependent and Text-Independent Verification

Verification is somewhat more complicated than identification, especially in a text-dependent context. The more specific details of speaker verification are discussed in this section.

2.3.1.3.1 Text-Dependent Speaker Verification

In a text-dependent speaker verification system, a user is required to say a specific password phrase that has been prompted by, and hence is known by, the system in order to be accepted. Not only does the system consider the unique speaker-discriminatory characteristics of the user's voice, but it must also determine whether or not the user

actually said the phrase the system prompted. This means that even if a user's voice matches one of the speaker models within the database, he or she will not be granted access to the system unless they have spoken the correct phrase, which is why the phrase is sometimes referred to as a "combination lock" phrase. Randomized phrase prompting, which is where the system randomly selects a password phrase each time it is accessed, is used to further enhance security. The underlying idea behind randomized phrase prompting is that it will make it more difficult for an imposter to gain false access to the system via an arbitrary tape-recording of a target speaker's voice, since the imposter will not be able to guess in advance what the prompt phrase will be. In addition, the specific idiosyncrasies of the person's voice are now tied to the context of a particular phrase. In other words, the unique characteristics of the person's voice that manifest themselves within the context of the lexical content within the phrase spoken are now used to determine similarity. Mimicry and falsification are more easily thwarted because random samples of the unique speaker characteristics of that voice are no longer adequate to gain access. These kinds of system are more secure than text-independent speaker verification systems for the reasons just mentioned.

An even more robust method is to collect a number of different phrases or a number of words, such as digits, at the time of enrollment of the target user that can be concatenated into new words and phrases at the time of verification. That way, an imposter cannot guess what the prompt phrase will be even if they somehow had access to the enrollment phrases beforehand. Unfortunately this improvement is not entirely bullet-proof either since there exist sophisticated electronic recording devices that can reproduce key words

in some requested order. Some of the speech segmentation methods used in Automatic Speech Recognition (ASR) may be useful for this purpose, especially if the system constructs new words and phrases based on old words and phrases. A description of some of these segmentation methods is given later on is this chapter in section 2.3.1.3.4.2. This may also be useful to simply try and align important corresponding speech events that are similar in two different utterances of the same phrase.

It is important to realize, however, that dissecting a set of utterances and then recombining them in new ways results in phrases in which phonemes and syllables will not always occur in natural ways, much like listening to the output produced by a Speak-and-Spell. This unnatural formation of new utterances results in information that is not necessarily characteristic of a given speaker, and thus contributes to increased error. The field of speech synthesis deals with such issues, in general.

A text-dependent recognition system requires the features that are extracted from the user's voice to contain both unique speaker-discriminatory characteristics and features containing information about the lexical structure of what was said. In this case, several speaker models are usually obtained for a single speaker. Each model represents the speaker saying a particular phrase, where each phrase is known to the system. When the system prompts an unknown user to say a phrase, it will pick a phrase that matches one of the speaker models for the target speaker. This model will then be used for classification.

2.3.1.3.2 Text-Independent Speaker Verification

In text-independent speaker verification, only speaker-discriminatory characteristics of the user's voice are considered, regardless of what has been said. In a text-independent system, any arbitrary tape-recording of the target speaker would allow them access into the system. While these systems are not as secure and constrained as text-dependent systems, it is more difficult to achieve a high accuracy from these. This is because there are no stipulations on what the user is saying, and so it is more important that the features that are extracted from the user's voice are independent of the lexical content of the speech. A larger range of vocal tract behaviors need to be modeled in order to adequately capture a complete model of the speaker's unique voice characteristics. Luckily, there are certain portions of the speech that we can exploit to this end. For example, regions of a speech signal that are more highly voiced, such as those areas associated with vowels, contain more highly speaker-discriminatory characteristics than regions of the speech signal that are less voiced, such as those areas associated with consonants. Some systems make use of a voicing detection algorithm (Campbell and Tremain, 1986), (Nemer and Goubran, 1997) to detect these regions of higher voicing, and only make use of these portions of the speech for recognition.

2.3.1.3.3 Speaker Modeling and Classification

A speaker recognition system is required to recognize certain target individuals. There are two major processes that must take place. First, a user of the system must be enrolled. Enrolment, which is known as the training phase, is the process of making a user known to the system. The system usually requires several utterances to be spoken, which are used to create a model of the user. Speech signal processing techniques are used to

extract highly speaker-discriminatory features. These features as a whole make up the parameters that are unique to that person.

In the testing phase, a user will be presented to the system, having made an identity claim. The system will require the user to say a particular phrase. The speaker model matching the claimed identity will be retrieved from the database of speaker models and a classification procedure will be performed to find out "how close" the given utterance is to the speaker model. A threshold measure is used to determine whether the utterance is "close enough" to the model to accept the user as being who they claim to be or not. If they are accepted, access to the system will be granted. If they are rejected, they will not gain access to the system. To do this, it must create one or more models of each person. The models are what represent the speaker, and are used to compare unknown speakers to see how well they match up. These comparisons are done in the classification stage.

Clearly, the choice of features, the method of representing the speaker model, and the choice of classifier is terribly important. Certain modeling methods work far better in combination with certain classification schemes, and work very poorly with other classification schemes. So, although a set of features may capture a speaker's unique voice characteristics very well, a poor method of comparison between the model and an input utterance from an unknown user may either fail to admit a valid user, or may fail to reject an invalid user. The worth of a classification system is normally judged based on the percentage of times it fails versus the percentage of times it passes the test correctly. Too many failures are unacceptable in a real-life application that will be used to gain

access to a secure system, such as the bank. This judgement call is made in different ways, depending on the purpose of the application.

Although the focus of this thesis is speaker identification, some of the difficulties in achieving high success rates in speaker recognition systems are more easily seen in a text-dependent context, which will be described next in a verification setting.

2.3.1.3.4 Text-Dependent Speaker Verification System

As already mentioned, the system usually requires several utterances to be spoken, which are used to create a model of the user. Speech signal processing techniques are used to extract highly speaker-discriminatory features. These features as a whole make up the parameters that are unique to that person. In the case of a text-dependent system, additional features are required that contain information about the lexical content of the utterances. The system must ensure that the speaker said what it told them to say. Figures 2.1 and 2.2 show the typical stages in the enrolment and verification procedures of a speaker verification system.

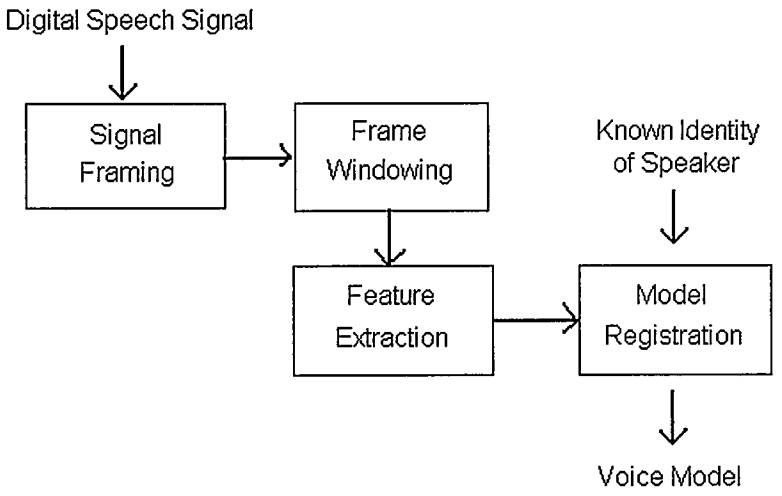


Figure 2.1: Enrolment process.

In the testing or verification phase, a user will be presented to the system, having made an identity claim. The system will require the user to say a particular phrase. The speaker model matching the claimed identity will be retrieved from the database of speaker models and a classification procedure will be performed to find out "how close" the given utterance is to the speaker model. A threshold measure is used to determine whether the utterance is "close enough" to the model to accept the user as being who they claim to be or not.

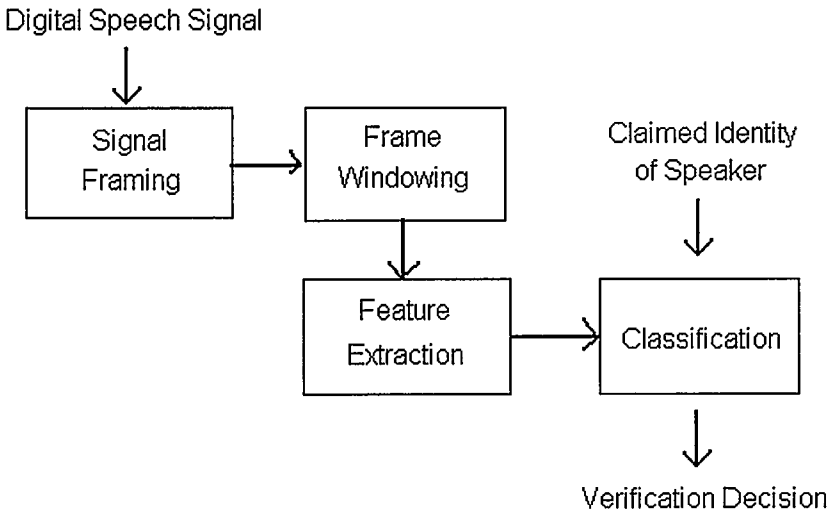


Figure 2.2: Verification Process.

There are two important computations that must be made in a verification system: False Accept Rate (FAR) and False Reject Rate (FRR). The FAR is a measure of the rate of the number of speakers that were accepted by the system when they shouldn't have been. When a person is falsely accepted by the system, this means that their identity did not really match the target speaker's identity in real life, but the system thought it did, and thus granted them access to the system as the target speaker. The FRR is a measure of the rate of the number of speakers that were rejected by the system when they shouldn't have been. When a person is falsely rejected by the system, this means that their identity matched the target speaker's identity in real life, but the system thought it didn't, and thus rejected them. It is highly desirable for a speaker verification system to have both a very low FAR and FRR, but in most cases it is clearly more important to place a stronger emphasis on a lower FAR, perhaps at the expense of an increased FRR. In general, if the

FAR is decreased, the FRR will increase, and vice versa. Depending on the application, this inverse relationship can be adjusted to suit the situation.

2.3.1.3.4.1 Verification Threshold

The verification threshold, which decides whether a distance or distortion measure is small enough (or "close" enough) to pass, is very important. There are several ways in which the verification threshold can be determined. In some systems, thresholds are hand-picked, and in others the thresholds are computed and updated as each new speaker enrols into the system. The latter is more robust because the threshold for every person is "tightened up" according to the properties of the other speaker models in the database, making the system all the more secure, at least internally. However, recalculation gets tedious as the database size increases, because the threshold must be recomputed for each person that has already been enrolled, as well as for the new person that is being enrolled.

2.3.1.3.4.2 Speech Segmentation

In a text-dependent speech recognition system, the lexical content of two different utterances must be compared to see if they match. However, no two utterances of the same phrase will be exactly alike, even if the utterances were made by the same person. Several factors, such as the rate at which someone is speaking and the manner in which they speak (for example, a person can whisper or sing and they can be angry, sad, or excited) can vary the timing and behavior of the speech events in the speech signals. In addition, factors such as accent and intonation can change the way in which speech is manifested. So, to perform matching between two speech signals, it is necessary to

somehow align like speech events. One way of doing this is to segment the utterance into smaller "molecules" or "atoms", such as phonemes, syllables, or words. These portions of speech can be used directly for recognition purposes or can be used as anchor points around which we can refine our analysis. Speech must be segmented into comparable units that can be reliably reproduced. The branch of speech analysis for which this is the main goal is Automatic Speech Recognition (ASR), which attempts to determine what has been said regardless of who said it (speaker-independency). This task can be extremely difficult, not only because of intraspeaker variability, but also because of interspeaker variability. Intraspeaker variability refers to the variability of pronunciation by one speaker. Shorter-term variability, such as coarticulation, is a big problem. Coarticulation is the pronunciation variation that is encountered in the speech of a single person and is explained in the section on segmentation of speech using the phoneme in this chapter. Also, if a person gets sick, the way their voice sounds changes temporarily. Longer-term variability, such as aging, must also be taken into account. A system must be able to adapt to the changes in a person's voice that occur as a result of getting older. Interspeaker variability includes differences in accent, intonation, gender, and age. Interspeaker variability is an asset in speaker recognition because it helps to separate one speaker from another.

Typically, frame analysis is first used to detect the endpoints of the selected speech units, such as phonemes, syllables, or words, at which time the variable-length anchor points can be placed for further analysis where necessary. Most systems that perform speech segmentation in the literature used fixed frame lengths and do not bother with a refined

variable-length analysis. This frame analysis is described in more detail in chapter 3. The following discussion helps explain why it is difficult to match the content of two different speech signals, even in the absence of noise and other extraneous factors.

2.3.1.3.4.2.1 The Phoneme

Traditionally, linguists have chosen the phoneme as the smallest speech unit. All speech sounds can be constructed using phonemes. Every different language has its own set of phonemes. The term "phoneme" usually refers to an individual speech sound that is specific to a given language, and the term "phone" refers to an individual speech sound independent of any language. The way in which a phoneme is produced depends on how the different places along the vocal tract are shaped and positioned, which is described in chapter 3.

Every utterance can be broken down and expressed in terms of its atomic set of phonemic units. The phoneme is a popular sub-word speech unit that has been widely used for many years (and is used still today) in segmentation schemes for ASR. However, in practise, machines still have an extremely difficult time matching the phoneme recognition accuracy that trained phoneticians are able to accomplish, without a lot of overhead and complexity. The phonemes defined in linguistics are canonical, whereas in practise, phonemes that are manifested in speech waveforms hardly ever conform to the canonical representation. This wide variation is due to many factors, including intonation, accent, pronunciation variation, and coarticulation. Coarticulation occurs when a phoneme's behavior is dictated by the neighboring phonemes surrounding it. The way a

phoneme sounds changes according to its context with respect to its neighboring phonemes. A phoneme that occurs in one context may sound entirely different, or may not be heard at all, in another context. For example, there are roughly 46 phonemes in the English language alone, but there are over 64,000 different phonemic contexts, not even taking into account the pronunciation variation that occurs across different speakers due to accent, regional dialect, etc, or the fact that a word or phrase is never spoken the same way twice by the same person. Clearly, trying to capture and model this vast amount of information is a difficult, and probably infeasible, task. Luckily, text-dependent speaker recognition does not require such a detailed representation of information. Only the data within a target key phrase or phrases are modeled, making the task somewhat easier.

An example of coarticulation can be seen when comparing the words "brother" and "elephant". The phoneme /ɛ/ in "el" at the beginning of the word elephant sounds different in the "er" in brother. The /ɛ/ assimilates with the liquid phoneme /r/ resulting in a change in the way it sounds based on the context in which it is found. Because of the sheer amount of pronunciation variation that arises due to coarticulation, even for a single person, the amount of training data required to capture and model this information is huge and usually infeasible, except for restricted tasks.

In recent years, alternatives to the phoneme as a speech unit for segmentation have been explored. Words have been revisited, and syllables have been explored. Diphones and triphones have also been popular recently. The properties and use of the syllable, which is made up of one or more phonemes, have been explored by researchers such as Steve

Greenberg (Greenberg and Kingsbury, 1997), (Shastri, Chang, and Greenberg, 1999), (Greenberg, 1998), (Kingsbury, Morgan, and Greenberg, 1998), Nelson Morgan (Kingsbury et al. 1998), Brian Kingsbury (Kingsbury et al. 1998), Hugo Meinedo, (Meinedo and Neto, 2000), (Meinedo, Neto, and Almeida, 1999), and others.

2.3.1.3.4.2.2 The Word

The word was one of the very first speech segmentation units, used by some very early researchers. However, it turned out to be a bad choice for a segmentation unit in the context of ASR since there are far too many words within a given language to have to train on to be able to adequately capture every possibility, although there are some who have used it more recently as a speech segmentation unit, such as (Stolcke and Shriberg, 1996), (Martens et al., 2002). This problem does not exist on the same scale for text-dependent speaker recognition because, once again, the number of words needed are limited to those within the target phrases that are modeled for a given speaker.

Some research that has been done on the segmentation of speech into word units brings to light a few very important issues regarding text-dependent speaker recognition in general. The word-segmentation approach was taken in (Higgins and Porter, 1991) for text-dependent speaker recognition. Rather than simply train speaker models on fixed, pre-trained phrases to be chosen randomly as a prompt to an unknown user of the system, Higgins et al. trained the speaker models on various words which formed a small vocabulary. A user would then be prompted to utter a phrase consisting of a concatenation of randomly chosen words from this vocabulary. They stated that although

there was benefit in having a small number of fixed phrases, and therefore a shorter time enrolment and high accuracy, the system was vulnerable because of its predictability. A larger set of phrases would provide more of the necessary randomness and high system accuracy, but it would mean an excessive enrolment time.

Most systems, however, cannot adequately achieve high success using the randomized phrase prompting strategy described above. According to (Higgins and Porter, 1991), "words occur in the test material in contexts that did not occur in the enrolment material. The context in which a word is spoken influences its pronunciation through coarticulation, caused by limitation in the movement of the speech articulators. These unmodeled coarticulations contribute to the measured dissimilarity between the input speech and the claimant's word templates, increasing the likelihood of rejecting valid users." (p. 89). Thus, as with phonemes, but to a lesser extent, word segmentation does not adequately address the issue of coarticulation directly.

It is important to understand that in ASR in particular, it was found that training on words was much too difficult since there were far too many words in one language alone to train them all and to be able to adequately capture all the necessary data to be able to recognize any and every word that could potentially be uttered by someone. In addition, as with phonemes, interspeaker variability further hindered these systems. It was used early on and abandoned for these reasons. However, in the context of an automatic speaker verification system, as already mentioned above, it is a more reasonable choice (apart from the randomized phrase prompting that Higgins was trying to achieve) since we have

a more limited set of possible words, and they are all being uttered by the same speaker. Thus, the problems associated with interspeaker pronunciation variability would actually aid a system such as this because these variations would help distinguish one person from another more readily.

2.3.1.3.4.2.3 The Syllable

Many have argued that the syllable is a better choice for speech segmentation than the phoneme. Steven Greenberg has done a good deal of in-depth research concerning the syllable. He discusses use of the syllable in speech segmentation in (Greenberg and Kingsbury, 1997), (Shastri et al., 1999), and (Greenberg, 1998).

In (Shastri et al., 1999) Greenberg discusses some of the reasons why the phoneme is a poor choice for speech segmentation, and why the syllable is a better alternative. In recent years, most ASR systems for the English language derive lexical information from a speech signal using phonetic segmentation. The automatic segmentation and labeling of an acoustic signal at the level of the phone is not very accurate, as compared to the segmentation and labeling done by trained phoneticians, due to coarticulation and variation in pronunciation. The performance of such systems deteriorate as they move out of a controlled environment and are put to use in a more realistic setting, fraught with a variety of environmental and linguistic conditions. He argues in (Greenberg, 1998) that little attempt has been made to provide an alternative lexical representation to the phone that is organized either above or below the level of the phone. However, it has been

shown that ASR systems based on the syllable, a more lexically stable model, are more accurate than systems using only phonetic segmentation.

To show why this is, Greenberg explains the anatomy of a syllable and addresses the problem of variation in pronunciation from an ASR perspective in (Greenberg, 1998). According to Greenberg, "In spontaneous speech, the phonetic realization often differs markedly from the canonical, phonological representation. Entire phoneme elements are frequently dropped or transformed into other phonetic segments". (p. 51). Greenberg suggests that while these phenomena appear complex and arbitrary at the level of the phoneme, the patterns of deletions and substitutions become systematic when placed within the framework of the syllable. The syllable can be dissected into three parts: the onset, the nucleus, and the coda, respectively. Most English syllables take on one of the following forms (where C stands for a singular or complex consonantal cluster and V stands for a singular or multiple adjacent vowels): CVC, VC, CV, V.

The onset of a syllable is typically consonantal, or made up of one or more consonants. It generally "survives" the changes brought about by coarticulation and varying pronunciation that afflict the nucleus and coda. The onset seems to dictate pronunciation of the rest of the syllable. It tends to approximate the canon, especially when it is complex, or made up of two or more consonantal segments, despite varying speaking conditions, and the more standard the articulation of the onset (or the higher the degree to which it approximates the canon), the higher the probability that the nucleus and coda will be also be pronounced in a canonical fashion. In general, the pronunciation of one

part of the syllable is affected by the pronunciation of the other parts. Greenberg suggests that this implies that the specific mechanism responsible for pronunciation looks beyond the individual phonological "atomic" building blocks to a higher level of control such as the level of the syllable, or even a level beyond the syllable.

The nucleus is vocalic, or made up of vowels. Nuclei are "chameleons" because they often deviate from the canonical and are thus capable of taking on a number of different vocalic appearances. This generally results in a substitution of one vocalic appearance with another, as opposed to a deletion altogether as is more common with the coda, since the deviation from the canon is likely to preserve the nucleus' vocalic form.

Finally, the coda is the portion of the syllable that is most likely to get disposed of by means of deletion or transformation into a segment that assimilates with and flows into that of the following syllable's onset. Unlike the onset, the complexity of the coda has no significant affect on the likelihood of a canonical pronunciation.

In light of this research, it seems reasonable to conclude that all three components are responsible for the phenomenon of coarticulation. The nucleus and coda make up the bulk of the coarticulation, and the onsets, both of the current syllable and the one following, are what dictate the manner of coarticulation. It follows, then, that the best way to perform syllable segmentation would be to train on and identify syllable onsets. Furthermore, there is evidence to suggest that auditory neurons are most likely to be responsive to the initial portion of a signal (Shastri et al., 1999). Arguments can be made

for training on the nucleus as opposed to the onset, as in (Mahadeva Prasanna, Gangashetty, and Yegnanarayana, 2001), which would, in fact, be better suited to speaker verification applications. As will be discussed later, the more sonorant portions of a speech signal, associated with the vowels, or nuclei, contain more speaker-discriminatory information, though perhaps at the expense of lexical content discrimination.

CHAPTER THREE: SPEECH SIGNAL PROCESSING

Analyzing speech in its digital form to extract meaningful data is comprised of several complex steps. A digital speech signal is first recorded and is then broken down into several segments called frames, and analyzed frame by frame using signal processing techniques. Signal processing allows a time-domain signal to be broken down into its frequency components, or bases, much like a coordinate in Euclidean 3-space can be broken down and represented in terms of the principle axes, which are the x-axis, the y-axis, and the z-axis (Gaswami and Chan, 1999). Once a signal is broken down into its frequency components, operations can be performed on these components to manipulate the signal and extract information that was otherwise hidden in the time-domain representation. The signal processing analysis yields meaningful features, which are used to model and classify a signal, depending on the application.

3.1 Speech Production

Speech is produced when air is pushed through the vocal tract, making contact with the various articulators that are shown in figure 3.1 (Campbell, 1997). As the speaker is saying a phrase, the shape of their vocal tract changes gradually over the duration of the utterance. The process of continually moving the different components of the vocal apparatus from one configuration to another results in the differing speech sounds that occur over the course of the utterance. The behavior of each articulator manifests itself in the corresponding portion of the speech signal. At the lips, which is the end of the line, a pressure wave is produced.

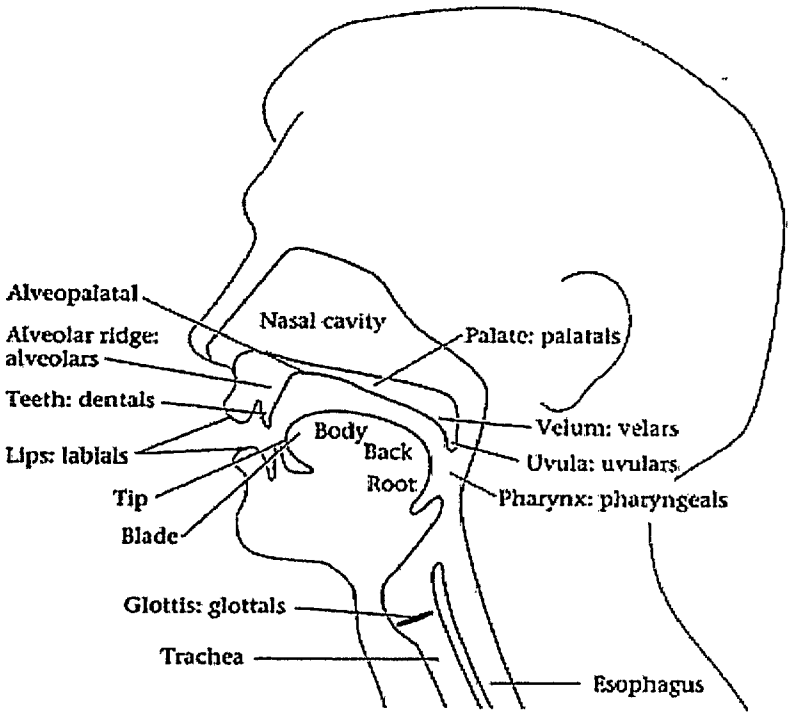


Figure 3.1: Places of articulation. (Campbell, 1997)

From this figure it is clear that there are many different structures involved in the production of speech. There are many complicated combinations of articulator positions and movements at different points along the vocal tract, which help produce the different sounds of human speech.

3.1.1 Voicing and Voicelessness

As air is pushed through the vocal cords, the position of the vocal folds makes a difference in the way a sound is manifested. When air passes through the vocal folds and they vibrate, the sounds are said to be voiced. This usually happens because the vocal folds are brought close together but are not completely closed. Voicing occurs in vowels

and some consonants. Vowels are sonorous, or highly voiced, syllabic sounds. Different vowel sounds are produced with different shaping and placement of the tongue and lips. When the vocal folds do not vibrate, the sounds produced are said to be voiceless. In this case, the vocal folds are positioned farther apart. Voicelessness occurs in consonants.

3.1.2 The Speech Signal

As previously mentioned, speech is produced when air is pushed through the vocal tract, producing a pressure wave at the lips. When performing speech analysis, this pressure wave is captured as a discrete digital signal by a recording device, such as a microphone. The continuous analog signal is converted to a digital signal through sampling, which can then be interpreted and manipulated by a computer. The speech signal, or waveform, is the machine's representation of the speech utterance, and can henceforth be broken down and analyzed.

Figure 3.2 (Gaswami and Chan, 1999) illustrates the digital sampling of a continuous signal function. A continuous signal is sampled by taking the value of the function at uniformly-spaced intervals in time. These discrete sample values are the digital representation of the original signal and are stored in a vector as a sequence of values.

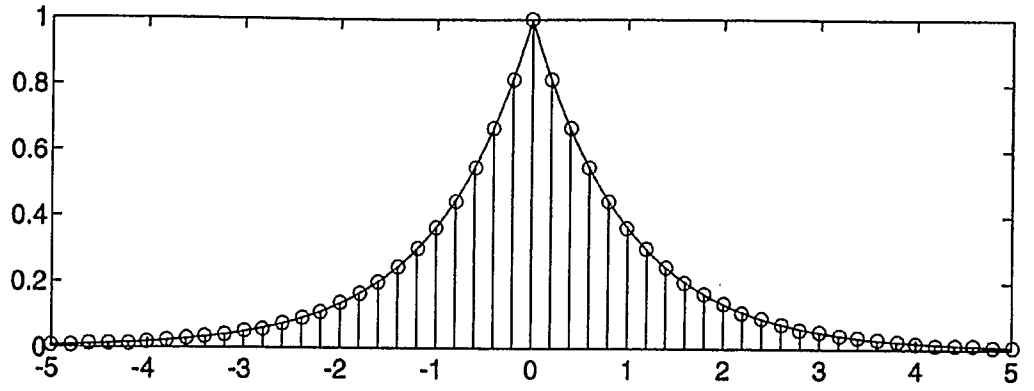


Figure 3.2: Sampling of a continuous function. (Gaswami and Chan, 1999)

As can be seen in figure 3.3 (http://www.ling.mq.edu.au/~rmannell/sph302/epg/sam.pdf), the changes in the configuration of the vocal tract across the utterance are reflected in the speech waveform itself. There are areas of greater excitation, or voicing, interchanged with areas of less excitation, including possible areas of silence and non-speech sound. In general, regions that are more voiced are found to contain a higher degree of speaker-discriminatory information than regions that are less voiced.

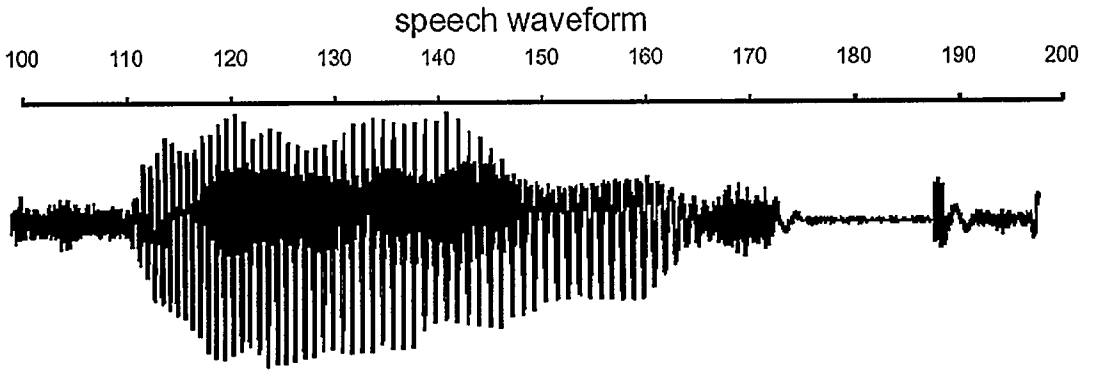


Figure 3.3: Speech waveform.
(http://www.ling.mq.edu.au/~rmannell/sph302/epg/sam.pdf)

Because of this gradual evolution of the vocal tract, the signal is said to be quasi-stationary, which means that it changes gradually over its course, or is slowly time-varying. In other words, the signal contains transient events that cannot be predicted even using past statistics. While the signal is doing one thing at time t, it is behaving completely in a completely different manner at time $t+n$. While the speech signal is globally non-stationary, or continually changing over its entire duration, it is locally stationary. This means that very small regions of the signal appear to be stationary, and are not changing. From a signal processing perspective, this has important implications. These implications will be explained later on in this chapter.

A speech signal, then, can be thought of as a sequence of piecewise functions, where each function $f_j(x)$ corresponds to a locally stationary region of the speech waveform:

$$f_1(x) + f_2(x) + \ldots + f_n(x) \tag{3.1}$$

These functions, as well as the regions of the speech waveform they correspond to, are unknown and cannot be predicted. No two utterances are exactly alike, even if they came from the same person. It is nearly impossible for a person to exactly repeat the same cycle of articulator movements that produced an utterance.

3.2 Speech Signal Processing

3.2.1 Fourier Analysis

Fourier analysis converts a signal from one domain to another domain. The original domain is referred to as the time or spatial domain, and the transform domain is referred to as the frequency or spectral domain. This transformation breaks a signal down into its frequency components. When a signal is transformed to the frequency domain, information that was otherwise hidden in the time domain is revealed, and operations can be performed on the frequency representation to manipulate or extract information from the signal. The Fourier method is one of the most powerful techniques available for signal analysis. In speech signal processing, the frequency domain is the domain from which features are extracted for the classification and pattern matching tasks described in chapters 2 and 4.

There are multiple methods of performing Fourier analysis. However, it is only possible to use one of these methods for speech signal processing, which consequently introduces complications in the analysis of the speech. These issues are discussed in what follows in this chapter. First, each method of Fourier analysis is explained, and a reason is given for why it cannot be used for analyzing speech signals. Each of the four Fourier methods are presented in pairs of equations, the first of which transforms a signal from the time domain to the frequency domain, and the last of which transforms from the frequency domain back to the time domain. Thus, the equations are inverses of each other. These equations, as well as an introduction to signal processing, are found in (Gaswami and Chan, 1999) and (Oppenheim and Schafer, 1989).

3.2.1.1 Continuous Fourier Analysis

Fourier analysis includes two methods of analysis, the Fourier series, and the Fourier transform. These methods are applied to analog signals (as opposed to digital signals), meaning that they are both defined for use with continuous, rather than discrete, functions.

The Fourier series is applicable to periodic signals and is given by:

$$p(t) = \sum_{k=-\infty}^{\infty} \alpha_k e^{jk\omega_0 t} \tag{3.2}$$

where the Fourier coefficients α_k are computed by:

$$\alpha_k = \frac{1}{T}\int_{t_0}^{t_0+T} p(t)e^{-jk\omega_0 t} \tag{3.3}$$

The α_k are the coefficients of the Fourier series, and the period is $T = 2\pi/\omega_0$. In addition, ω_0 is the fundamental frequency. The fundamental frequency is usually the strongest frequency that appears in the sound, and seems to most closely indicate the perceived pitch of the total sound. Note that equation 3.2 takes a signal from the time domain to the frequency domain, and equation 3.3 is the inverse, which takes the frequency domain back to the time domain.

The Fourier transform is an extension of the Fourier series. It can be applied to aperiodic functions that are defined on the real line by allowing the period T of a periodic function to extend to infinity. It is given by:

$$\hat{p}(\omega) = \int_{-\infty}^{\infty} p(t')e^{-j\omega t'}dt' \tag{3.4}$$

and its inverse to get back to the time domain is:

$$p(t) = \frac{1}{2\pi}\int_{-\infty}^{\infty} \hat{p}(\omega)e^{j\omega t}d\omega \tag{3.5}$$

Once again, equations 3.4 and 3.5 are the inverse of each other. Clearly, the computation of the Fourier series coefficients α_k and the Fourier transform requires integration, and therefore the signal must be represented as an analytic function $f(x)$ that can be integrated. However, most of the signals that we encounter in real life, including speech signals, cannot be represented as such. No signal $f(x)$ is known for a given input speech signal, and none can be predicted.

3.2.1.2 Discrete Fourier Analysis

In cases where continuous Fourier analysis is not applicable, it is possible to convert the signal from analog to digital by sampling the original signal and obtaining a discretized signal. In digital form, the discrete-time Fourier series and the discrete-time Fourier

transform can be computed directly and used to obtain a frequency domain representation. These produce an approximate spectrum of the original analog signal.

The Discrete-Time Fourier series is given by:

$$f_p(n) = \sum_{k=0}^{N-1} \alpha_k e^{(j2\pi/N)kn} \tag{3.6}$$

$$\alpha_k = \frac{1}{N} \sum_{k=0}^{N-1} f_p(n) e^{(-j2\pi K/N)n} \tag{3.7}$$

and the Discrete-Time Fourier transform is as follows:

$$\hat{f}(\omega) = \sum_{n=-\infty}^{\infty} f(n) e^{-jn\omega} \tag{3.8}$$

$$f(n) = \int_{-\pi}^{\pi} \hat{f}(\omega) e^{jn\omega} d\omega \tag{3.9}$$

If a discrete signal is aperiodic, it can be considered to be a periodic signal with period $N = \omega$. In this case we extend the discrete Fourier series analysis to the discrete-time Fourier transform (DTFT), similar to the extension in the analog domain. In DTFT, only the time variable n is discretized. The frequency variable ω remains continuous. To

transform from the time domain to the frequency domain, a summation is taken, but to get back to the time domain from the frequency domain, an integral must be calculated.

Unfortunately, the DTFT can only be applied to signals that are infinite in length, or more specifically, have an infinite number of input samples, which we rarely encounter in real life, such as speech signals in particular. A speech signal will always have a finite number of samples, since even the original analog signal is finite in length.

The only remaining choice is the Discrete Fourier Transform. It is a version of the DTFT that acts on finite-length time and frequency domain representations. In the DFT, the frequency domain is sampled to produce a discretized spectrum. It is thus the equivalent to the set of equations (3.6 and 3.7) given for the Discrete-Time Fourier Series. Thus, for speech signals, the Discrete Fourier Transform (DFT) must be used.

3.2.2 Short-Time Frequency Analysis (Time-Frequency Analysis)

As already mentioned, while the speech signal is globally non-stationary, it is locally stationary, which means that the local spectrum corresponding to each change in the speech as a result of the change in the vocal tract is different from other parts of the signal. It is necessary to analyze each of these local regions of the speech, since they contain information about the behavior of the vocal tract and thus information about the characteristics of the lexical content and speaker's voice characteristics at that point in time.

For example, in speaker recognition, the unique characteristics of a person's voice are exhibited in different ways across the speech signal. Regions of voicing contain different information than regions that are not voiced. Furthermore, not all regions of voicing (or non-voicing) are alike. Phonemes, which are the basic building blocks of speech, or syllables, make up multiple different speech sounds, each of which contains its own set of information about the person's voice characteristics. Therefore, it is necessary to analyze each local speech sound or region in order to adequately capture all the necessary information about a person's voice that is being exhibited. The same is true in speech recognition, in which multiple speech sounds are combined to form a speech utterance, containing information about what the person said, and this speech information is distributed across the speech signal in differing ways.

Since it is necessary to extract information out of local regions of speech, signal processing cannot be performed over the entire waveform all at once. The reason for this is as follows. The Fourier spectrum only contains frequency-domain information, and does not contain any time-domain information. This becomes problematic because in complex signals, it is impossible to tell, by just looking at the frequency-domain, what part of the original time-domain signal is responsible for producing what characteristics in the frequency-domain. This is illustrated with the following example, taken from (Gaswami and Chan, 1999).

In the top part of figure 3.4 (Gaswami and Chan, 1999) is a time-domain function. It is a truncated sinusoid with a frequency of 4Hz. Note that there are spikes at around $t = 0.7$ and $t = 1.3$ seconds. The bottom of the figure shows the frequency-domain representation of this sinusoid. The dominant spike in the frequency-domain is due *primarily* to the sinusoid, while the smaller ripples along the frequency axis are due *primarily* to the perturbations, or delta functions (which are sharp changes) in the time-domain.

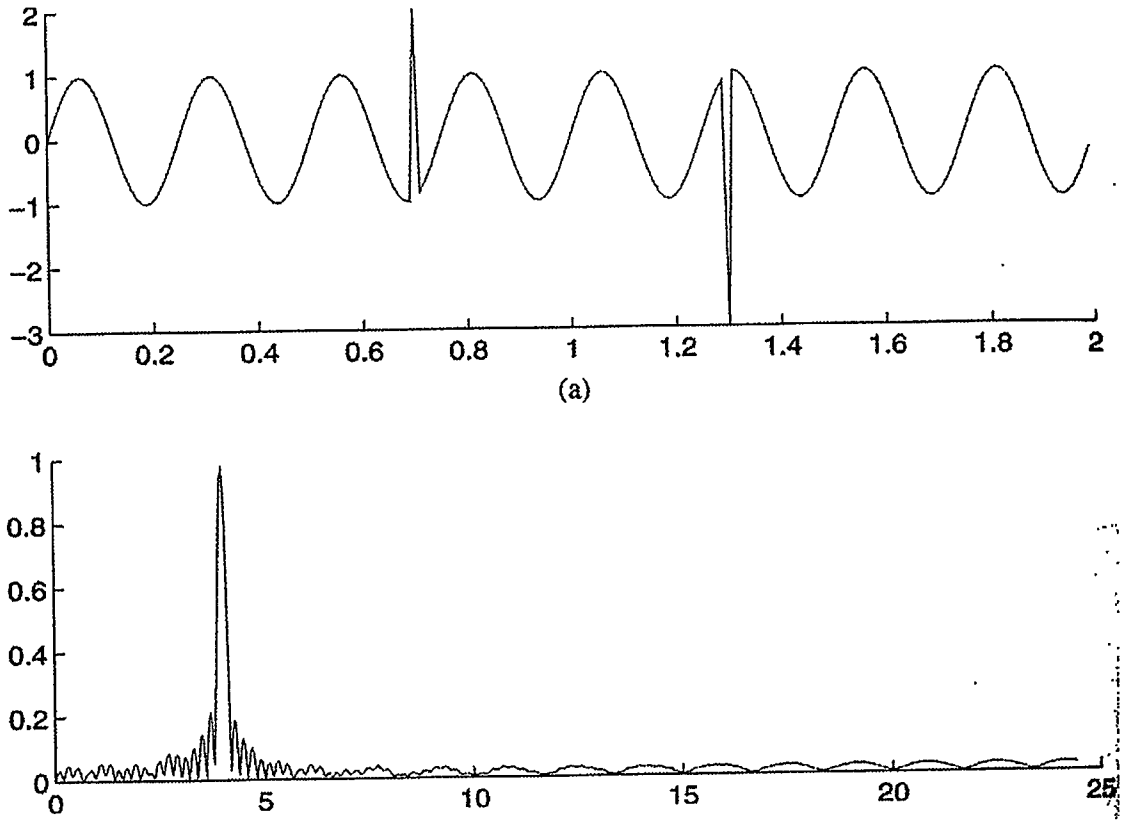


Figure 3.4: (Above) Sinusoid with spikes at 0.7 and 1.3 seconds. (Below) Frequency spectrum of the above sinusoid. (Gaswami and Chan, 1999)

If only the frequency-domain representation were considered, it would be impossible, from this information alone, to point out the locations at which the time-domain delta

spikes occurred, even though it is clear that there is evidence that they are there somewhere. So, analyzing the frequency information of a small region of the time-domain, which is what is required in speech analysis, is impossible when performed on a frequency representation of the entire time-domain because the frequency components of the entire signal as a whole are mixed together. The time information is lost.

If a Fourier analysis were performed over a spatial signal that varies too greatly, the analysis would not adequately characterize the details of the fine harmonics of localized areas of the waveform, and important information would be lost or obfuscated. The spectra of successive vocal tract changes would be intermixed. Thus, it is necessary to perform a local analysis, referred to as a short-time Fourier transform, in order to combine the time-domain and frequency-domain information. This can be done by breaking down the signal into smaller frames of analysis to capture this local spectral information, where each frame is analyzed separately.

3.2.3 Frame Analysis

To perform a short-time frequency analysis on the speech signal, it is reasonable to break the signal up into several short, successive segments, or frames, and analyze each frame individually and independently, each containing its own segment of the signal. This procedure is known as frame analysis. The length of the segments can be fixed or varying. In applications with varying frame lengths, the varying frame lengths are usually used once a preliminary fixed-frame analysis has been performed to detect important time events. Once these time events have been located, they can be used as anchor points

around which a variable-length frame can be constructed for further analysis of the new segments of interest.

The choice of the length of a frame is important. If the frame is too large, it will suffer from the problems described above, in that it will not be possible to determine which portions of the time-domain that the frequency-domain behavior corresponds to. If the frame is shorter than the ideal length, we can still obtain reasonable spectral results from a frequency analysis. However, if a frame is too short, it will not contain enough information to be of any use. The frame must be long enough to capture meaningful data. In speech applications, frame lengths are usually 10 to 20 ms long. These lengths are apparently not long enough to cause the problems that frame analysis and short-time frequency analysis are meant to avoid, and are still long enough to capture interesting and useful information. Figure 3.5 (Hermansky, 1990) depicts the framing of a speech signal.

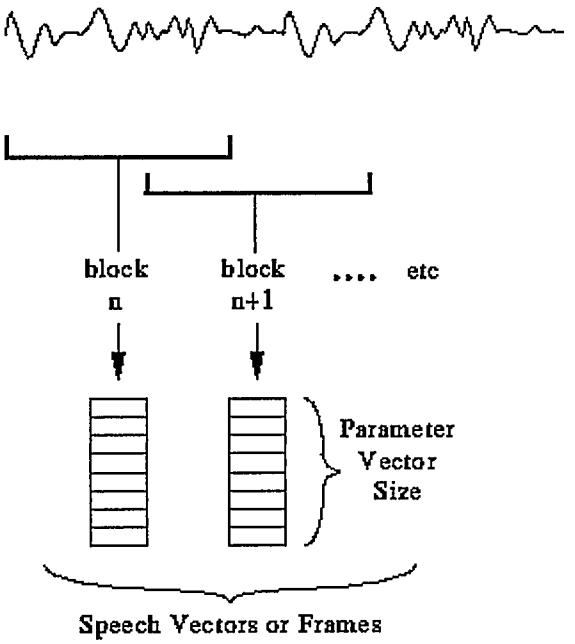


Figure 3.5: Frame analysis. (Hermansky, 1990)

3.2.4 Frame Windowing

3.2.4.1 Gibbs Phenomenon

Consider a square wave function $f(x)$, as shown in figure 3.6. The square wave travels along at 1 for awhile, then makes a sharp transition to -1 and continues at -1. The sharp corners or transitions made from 1 to -1 are discontinuities in the function.

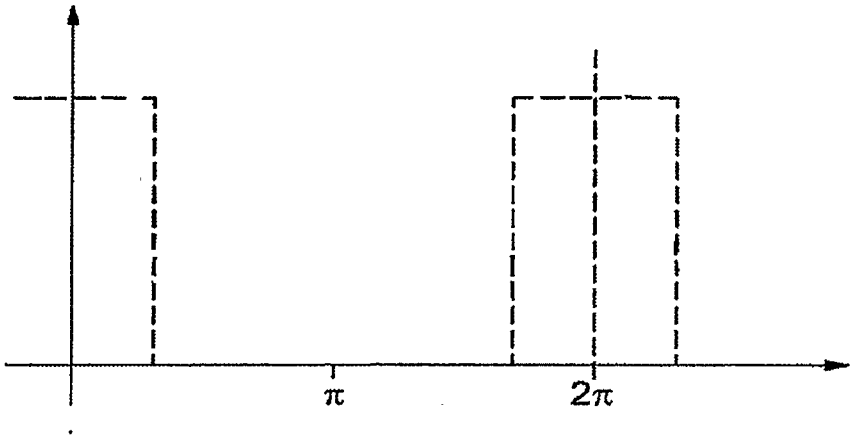
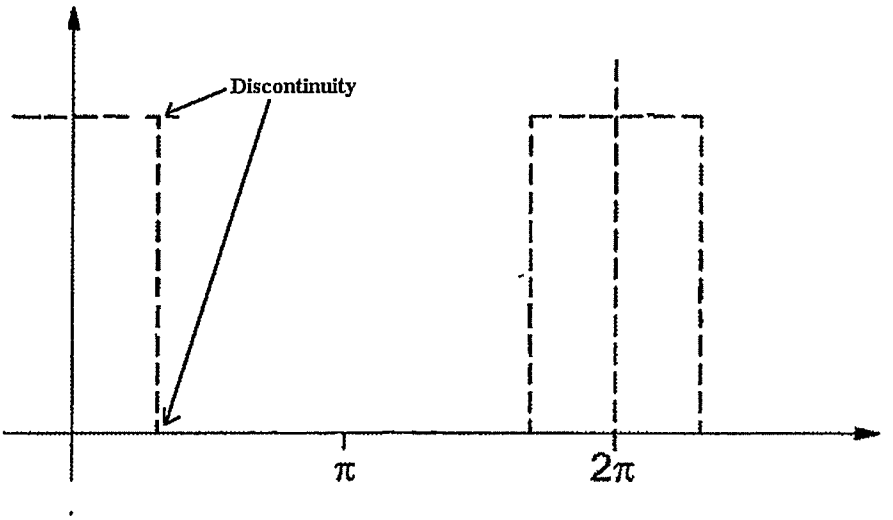
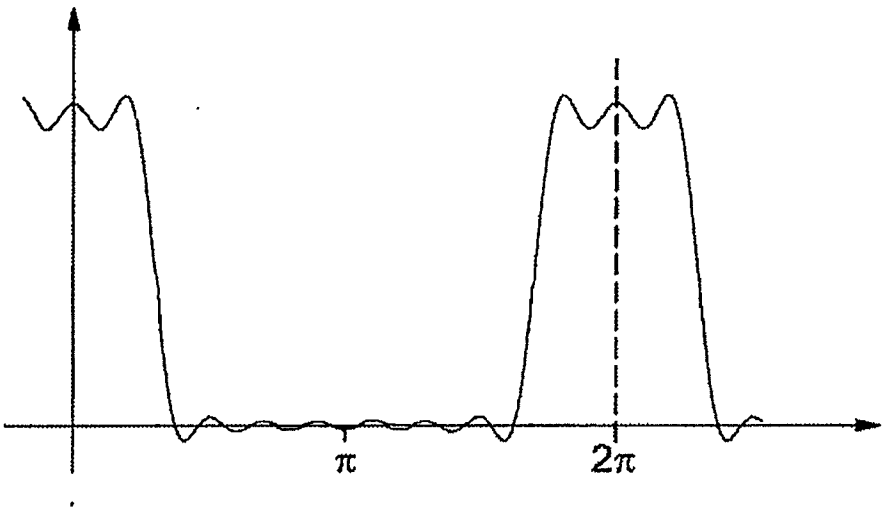


Figure 3.6: Square wave.

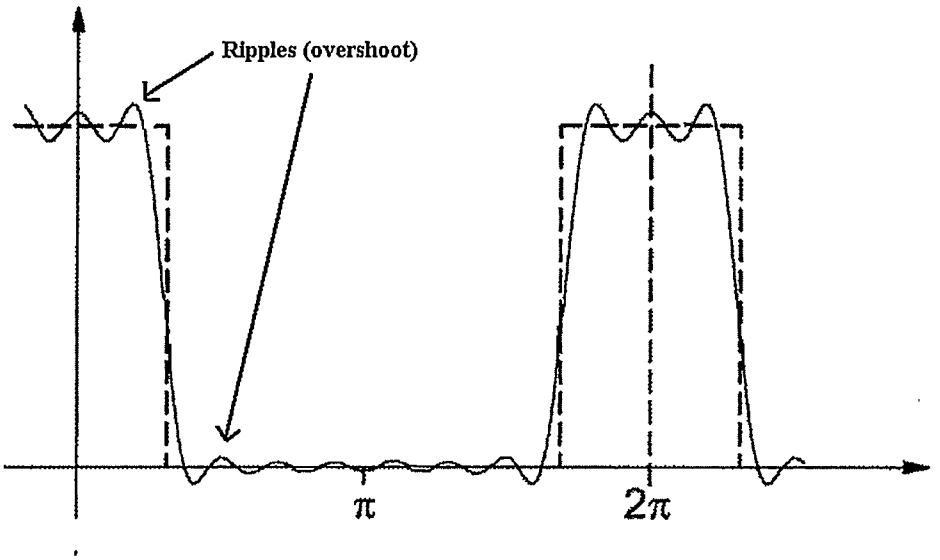
Suppose the fourier transform $f'(x)$ is taken on $f(x)$. Now suppose we want to reconstruct the original signal $f(x)$ from $f'(x)$ by taking the inverse fourier transform. However, what we end up with is a function $g(x)$ that is not equal to $f(x)$, as shown in figure 3.7. $g(x)$ contains ripples, caused by what is called "ringing", or a sharp overshoot, in the frequency domain. These ripples are magnified in places where the sharp discontinuities in $f(x)$ occurred. This is known in signal processing as the Gibbs Phenomenon. In figure 3.7, part (a) shows the original signal $f(x)$, part (b) shows the reconstructed function $g(x)$, and part (c) shows the g(x) plotted over top of $f(x)$ to make the ripples and artifacts more readily apparent.



(a) Square wave discontinuities in $f(x)$.



(b) Reconstruction of the square wave $g(x)$ from fourier representation $f'(x)$.



(c) Reconstructed function $g(x)$ overlaid on original function $f(x)$.

Figure 3.7: (a), (b), and (c) depict Gibb's phenomenon at signal discontinuities.

The more the number of terms K used in the Fourier series calculation, the more like the original signal the approximation becomes. As shown in figure 3.8, increasing the number of K terms decreases the severity of the ripples in $g(x)$, but even when the number of terms is extended to infinity, the ringing, and subsequently the ripples, never quite disappear (http://cnx.rice.edu/content/m10687/latest/). As shown in figure 3.8 (http://cnx.rice.edu/content/m10687/latest/), the ripples get narrower but do not get shorter. Thus, as the number of coefficients K approaches infinity, the reconstructed signal is the same as the original except at discontinuities.

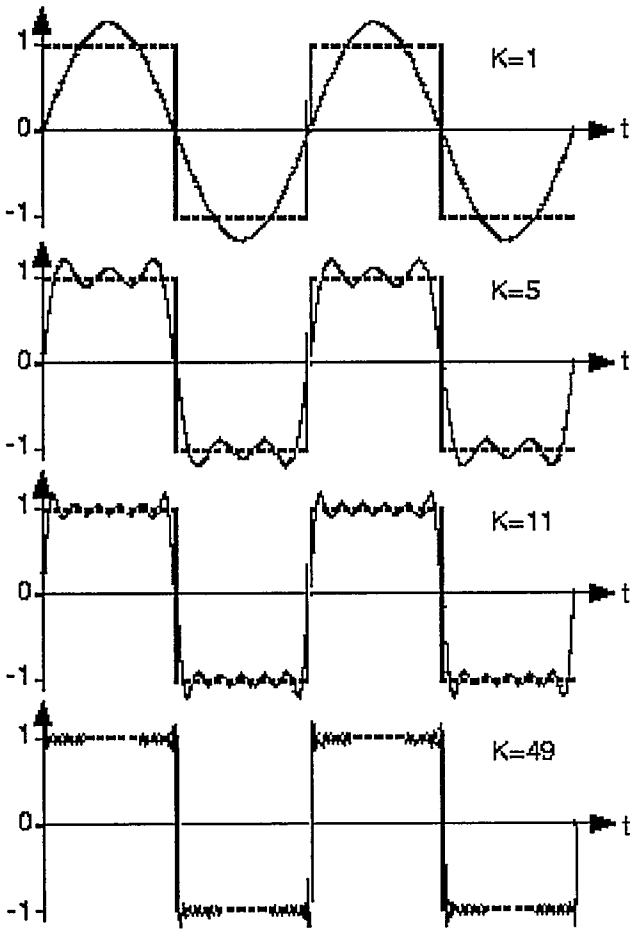


Figure 3.8: Fourier series approximations to the square wave using different number of Fourier coefficients K. (http://cnx.rice.edu/content/m10687/latest/)

This behavior is highly undesirable. Clearly, $g(x)$ contains data, or artifacts, that were not present in the original signal $f(x)$. Thus, using $f'(x)$ to perform any kind of feature extraction of the signal, where $f'(x)$ really represents $g(x)$ and not $f(x)$, will mean analyzing information that is not representative of the true input signal, and therefore the results will be unexpected and inaccurate.

3.2.4.2 Framing and the Gibbs Phenomenon

Now, consider a frame of a speech signal, $f(x)$, shown in figure 3.9. It is highly unlikely that $f(x)$ will be periodic, so we will assume that it is aperiodic. However, when performing a discrete fourier transform on the signal $f(x)$, it is assumed that $f(x)$ is really periodic, as shown in figure 3.10.

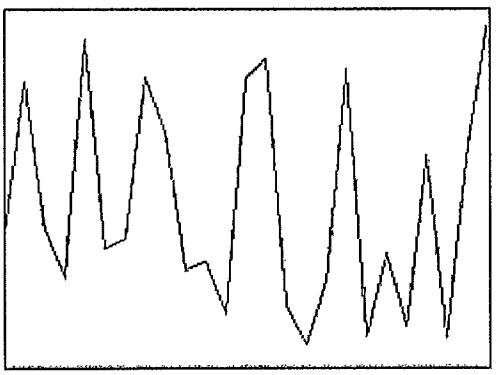


Figure 3.9: Frame containing f(x)

Since $f(x)$ is really aperiodic, sharp discontinuities appear everywhere one period meets the next. In figure 3.10, two periods of a framed signal have been placed side by side. Where they meet the signal has a jump discontinuity, meaning that it does not make a smooth transition into the next period. Due to the Gibbs Phenomenon, this will result in a fourier representation of a signal that contains artifacts that are not really there, which will give us feature vectors that represent something other than what was intended.

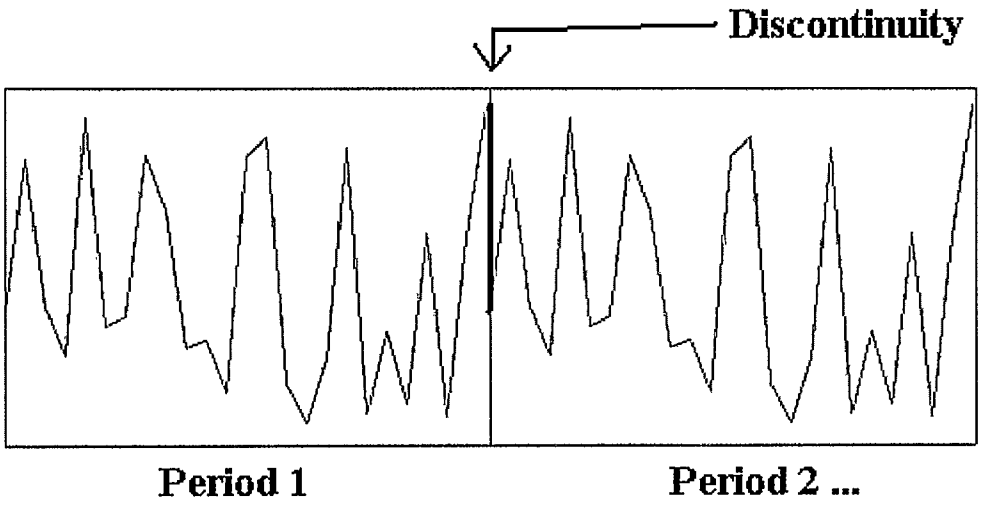


Figure 3.10: f(x) represented as being periodic. Two periods of f(x) are shown. A sharp discontinuity occurs where period 1 meets period 2. This will result in the Gibb's phenomenon occurring in the frequency domain.

The solution is an operation called "windowing". Windowing is necessary to taper off the signal due to the discontinuities that occur at the endpoints of the signal segment, making it appear periodic and removing the ringing that otherwise would have occurred.

As shown in figure 3.11, the portion of the signal within a frame is multiplied with a window function in the time domain to produce the resulting windowed signal. Typically a Hamming window is used for this, which is defined by:

$$w(n) = s(n)\left[0.54 - 0.46\cos\left(\frac{2\pi n}{N}\right)\right]$$

where $s(n)$ is the signal at time n and N is the length, or number of samples, of the signal.

The resulting signal is tapered off at the endpoints. Since the window is concentrated in a narrow band of frequencies around the middle of the frequency spectrum at $\omega = 0$, the frequency response of the windowed signal will still behave in a similar manner to the frequency response of the original unwindowed signal, but now the discontinuities have been removed and signal processing analysis can be performed to obtain the local spectral information of the signal segment contained within the frame.

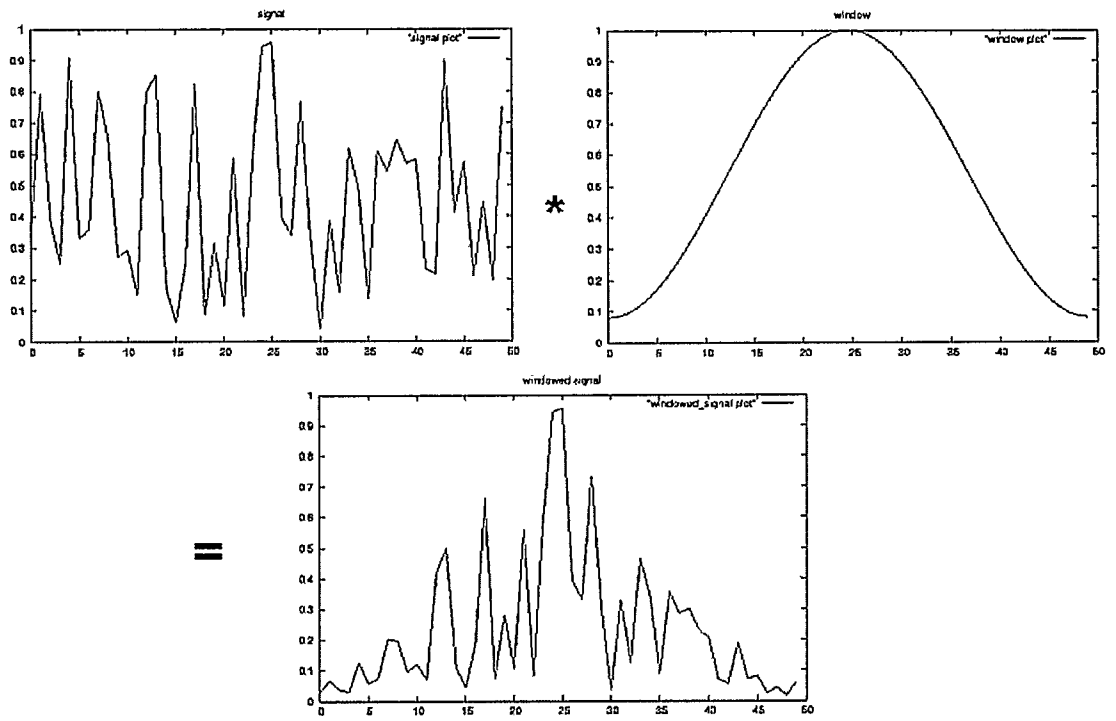


Figure 3.11: Windowing of a signal. A signal (top left) is multiplied with a Hamming window (top right) to produce a windowed signal (bottom).

Applying a window to $f(x)$ yields a windowed version $h(x)$, in which the right and left portions of the signal within the frame are "tapered off", while the portion of the signal in the middle of the frame is roughly what it used to be.

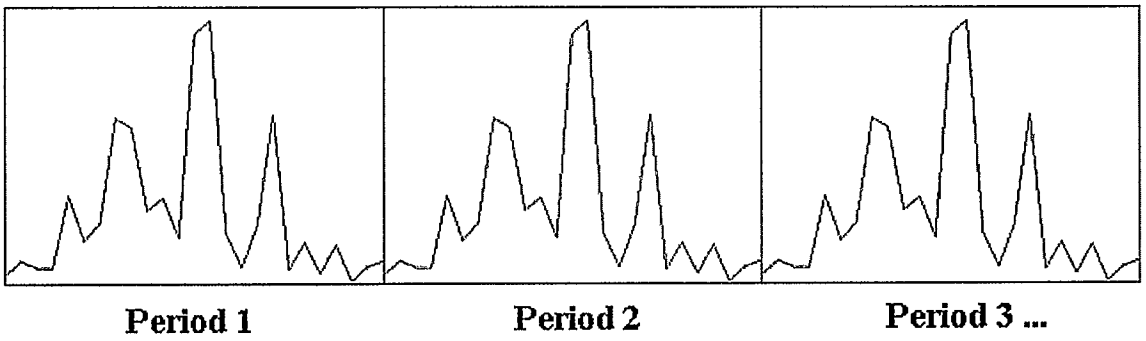


Figure 3.12: f(x) has been windowed to obtain h(x). Three periods of h(x) are shown. Sharp discontinuities between adjacent periods are now minimal.

Now, $h(x)$ is no longer aperiodic but is periodic. Thus, when the discrete fourier transform is performed on $h(x)$, it actually represents $h(x)$ and not some signal $p(x)$ that contains introduced artifacts.

3.2.5 Frame Overlapping

Figure 3.13 shows two adjacent frames of the original signal after they have been windowed. The two frames are now in an appropriate state for fourier analysis. There is still a problem with the analysis of $f_1(x)$ and $f_2(x)$, however. While we avoided introducing information that was not there (due to the Gibbs Phenomenon) by windowing $f_1(x)$ and $f_2(x)$, we have effectively lost information due to the windowing operation. The information that existed where the first frame makes a transition into the next is largely lost due to the tapering effect of the window function. This is especially

pronounced where $f_1(x)$ meets $f_2(x)$. The tapering effects of the windowing operation are especially strong at the beginning and end of a given frame. Therefore, where two frames meet, there is hardly any information left at all.

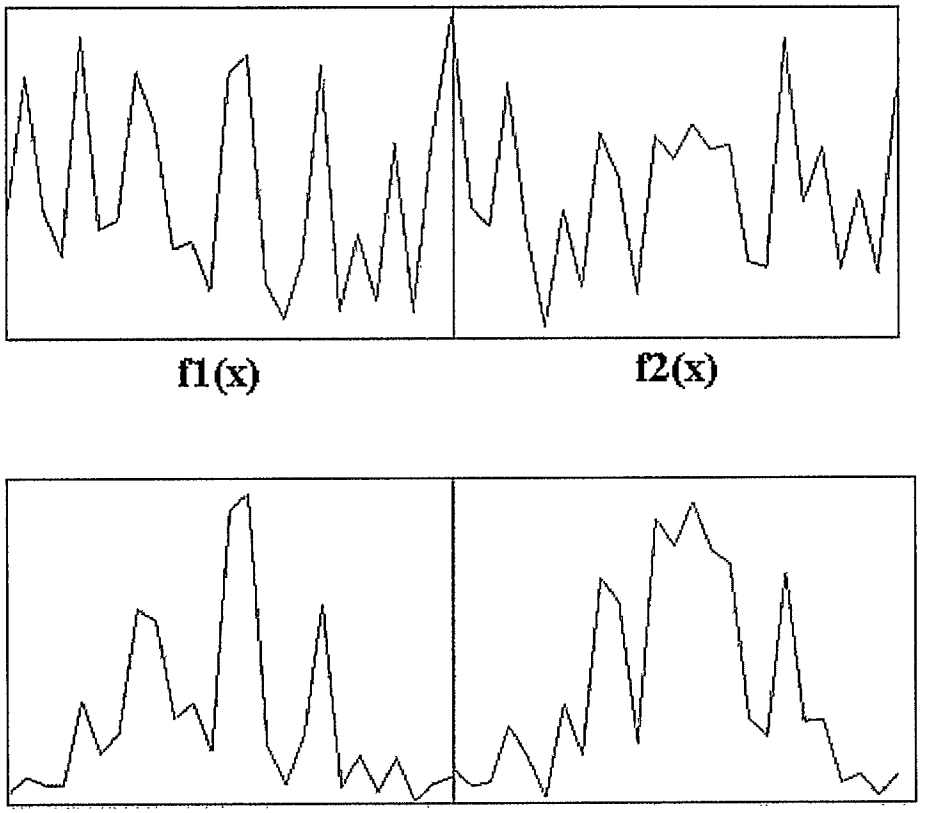


Figure 3.13: (Above) Two adjacent frames, $f_1(x)$ and $f_2(x)$. (Below) The same two frames after windowing.

To account for this, a third frame is introduced to capture $f_3(x)$, which straddles $f_1(x)$ and $f_2(x)$ as shown in figure 3.14.

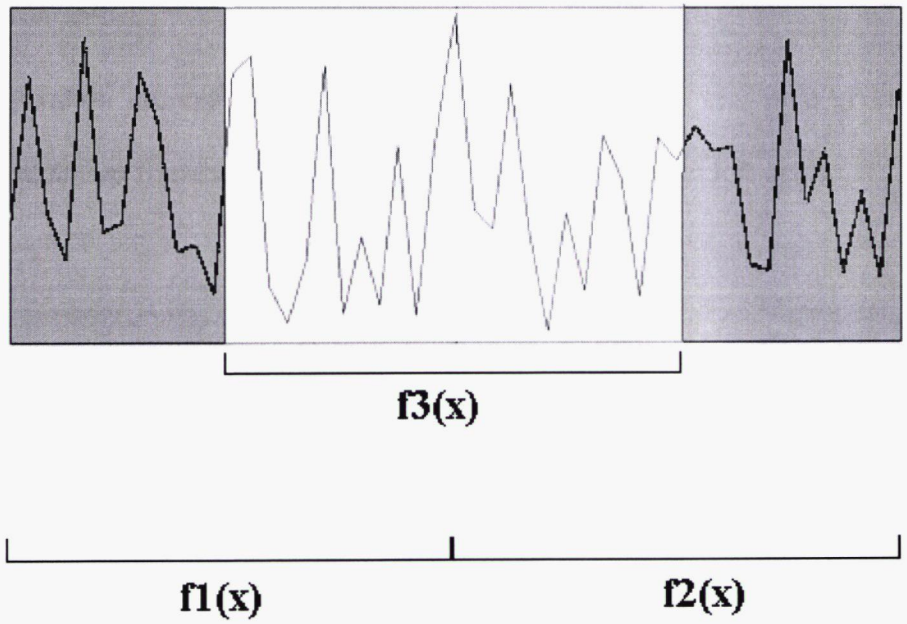


Figure 3.14: Overlapping frames

This places $f_3(x)$ in a position such that the information that was lost where $f_1(x)$ and $f_2(x)$ met will be most strongly represented (i.e. - in the middle of the frame) after the windowing operation has taken place.

In this way, the overlapping frames capture whatever information is lost between two adjacent frames, and all of the information that was present in original signal $f(x)$ is taken into account. A frame that overlaps two adjacent frames will then capture the interframe information that was otherwise lost in the transition from one frame to the next.

CHAPTER FOUR: SPEECH MODELING

Speech and speaker recognition have been around for several decades. Many of the techniques for extracting the desired characteristics of speech that were used 30 years ago or more are still in common use today, with slight variations. While more robust methods of recognition have come about by combining several different feature vectors into one feature vector, most of the advancements in speech and speaker recognition have come about by moving from template speech and speaker models to increasingly complex hybrid statistical methods of modeling speech.

This chapter is organized as follows. First, the issue of background noise is briefly addressed. Then, an overview of the main features explored in this thesis is given. Finally, a brief discussion of some of the classifiers and recognition systems that are commonly used for speaker recognition is given. Features and classifiers are discussed separately because there are so many different combinations of features and classifiers in the literature.

4.1 Noise

Like most signal and image processing operations, voice analysis is often subject to noisy input data. Artifacts that are extraneous to the task at hand become integrated into the speech data at the time of recording. In other words, there will never be a case when only the target person's voice is present in a voice recording. Noise comes in many different forms, from interference from the recording device, creating static or white noise, to more

natural background sounds such as other people's voices, traffic, birds chirping, a baby crying, or a cough or sneeze can be considered noise. It is an extremely difficult task for a computer to be able to separate all the noise from the target speaker, and in most cases, it is impossible. If noise is present at the time of training, it will be erroneously considered to be a characteristic of the speech content or the speaker's voice. When testing, any input data that would otherwise be correctly classified could be rejected because this same noise may not be present in the test data. In other words, what should be modeled isn't being modeled correctly. Similarly, when test data contains noise, what should be tested isn't being tested correctly.

The statistical modeling methods that are described later are more robust under noise than non-statistical methods because they allow for slight variations in the data that may be due to noise. While there is much research on the subject of dealing with noise in speech analysis directly, it is not within the scope of this thesis, and will only be referred to briefly in relation to the statistical modeling methods that are relevant to the material herein.

4.2 Speech Characteristics

Feature extraction is performed for several reasons. First, it greatly reduces the dimensionality of the problem. A frame of speech that contains 320 samples may have a set of features representing this frame having only have 12 or 13 coefficients. Thus a computationally infeasible problem becomes possible and much more realistic. Second, only the characteristics that apply to the task at hand are used in the analysis. For

example, speech data contains information about both who is speaking and what is being said. If the task is speech recognition, we would ideally like to remove the speaker-specific information, leaving only the speech content information.

Interestingly, many of the techniques used in speech recognition are also used in speaker recognition. The reason for this is because in most features, and despite much effort to the contrary, there is both speech content and speaker-specific information present. Ideally, in most situations, the two would be decoupled entirely. However, this is not entirely possible. There is still much that is not known about the complexities of speech, and many of the fundamental assumptions that are made about how the process of air being pushed through the vocal tract produces speech sounds are incorrect or limited.

4.2.1 Linear Prediction (LP) Coefficients

Linear prediction (LP) (Makhoul, 1996), (Makhoul, 1975) is an autoregressive method of feature extraction. This means that what the speech is doing at time t is determined by what the speech was doing at times $t-N$ to $t-1$, so the previous behavior of the speech is used to predict the future behavior. It has the effect of smoothing the spectral envelope of the speech. In other words, the complex local fluctuations of the input speech waveform are somewhat smoothed away. This is somewhat desirable because some of these fluctuations are probably due to noise.

Linear Prediction is one of the most powerful and important methods of deconvolution and parameterization of the source and filter. The source, or excitation, is the driving

force of speech production. It is the air that is pushed from the lungs through the vocal tract. The filter is the vocal tract itself. As the air is pushed through, it changes the air pressure that is created by the force of the air. By the time the air is expelled from the lips, the air pressure creates waves of sound that reflect the vocal tract changes that took place. LPC is explained in detail in (Campbell, 1997) and (Hermansky, 1990). It is one of the most commonly used features in the literature. It is often either used directly or is the basis of further feature extraction (http://www.isip.msstate.edu/publications/courses/ece_8463/ lectures/current/lecture_16/lecture_16.pdf). In fact, the autoregressive idea is used in a slightly different way in many psychoacoustic features, which are discussed later on in this chapter.

Linear prediction is based on the idea that the vocal tract can be modeled by a series of nonuniform, piecewise acoustic tubes that are joined together.

Figure 4.1 (Campbell, 1997) shows a crude acoustic tube model that is used to model the vocal tract. Adjacent sections of the tube vary in shape and diameter. The excitation signal, or source, is the driving force of speech production. It passes through the vocal tract, or filter, from left to right, to produce a speech sound $s(n)$. Most speech models, including linear prediction, attempt to decouple the source from the filter as a preliminary step.

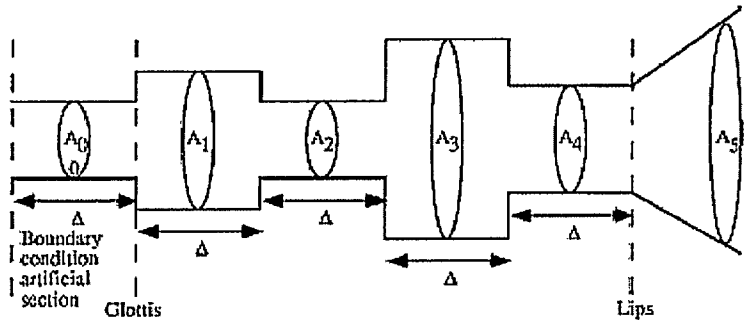


Figure 4.1: Acoustic tube model. (Campbell, 1997)

The all-pole LP models a signal $s(n)$ by a linear combination of its past values and a scaled present input. In the time domain, it is as follows:

$$s(n) = -\sum (a_k * s(n-k) + G*(u_n) \tag{4.1}$$

where $s(n)$ is the present output, α_k are the prediction coefficients, $s(n-k)$ are the past outputs, G is a gain scaling factor, and u_n is the present input, which corresponds to the human vocal tract excitation. This is simplified further because in speech applications, only the vocal tract, or filter, is kept, and therefore the input u_n is generally removed since it is unknown. Thus we get equation 4.2, which now depends only on past outputs:

$$\hat{s}(n) = -\sum (a_k * s(n-k)) \tag{4.2}$$

Now the problem of estimating α_k is easier because the source and the filter have been decoupled. The source, u_n, is not modeled by these prediction coefficients, and thus, by ignoring it, it is probably reasonable to assume that some sort of valuable speaker-dependent information that is present in the excitation signal has been lost. The linear prediction coefficients are typically found using a mean-square estimate. It has been found that minimizing the error signal $e(n)$ in this way produces a flat, or band-limited white magnitude spectrum of the error signal, which can be defined as being the difference between the actual signal $s(n)$ and the estimated reconstruction of the signal using the prediction coefficients, $s(n)$:

$$e(n) = s(n) - \hat{s}(n) = s(n) + \sum (a_k * s(n-k)) \tag{4.3}$$

Using $s(n)$ above, the LP Z-domain transfer function is as follows:

$$H(z) = \frac{G}{1 + \sum (a_k * z^{-k})} = \frac{G}{A(z)} \tag{4.4}$$

where $A(z)$ is known as the inverse filter. $A(z)$ is used to derive a set of features called Line Spectrum Pair frequencies, which is described in section 4.2.2.

One criticism that has been made about LPC is that the underlying assumption about the piecewise acoustic tube is incorrect, or inadequate. For example, the acoustic tube model

is static, whereas in reality, the human vocal tract is always changing shape. However, the beauty of LPC is in its simplicity, especially when unknown terms such as u_n can simply be factored out of the equation altogether.

4.2.2 Line Spectrum Pair (LSP) Frequencies

Line Spectrum Pair (LSP) frequencies are an alternative spectral representation of the linear prediction coefficients, and are discussed in (Soong and Juang, 1984). In (Soong and Juang, 1984) the use of LSP frequencies in speech data compression is discussed and some of their main properties are described and proven. An explanation of Line Spectrum Pair frequencies is also given in (Itakura, 1975).

Line spectrum pairs are essentially a representation of p^{th}-order coefficients of the inverse Z-domain filter $A(z)$ of the LP all-pole representation as follows:

Let

$$A(z) = \frac{1}{2}[P(z) + Q(z)] \tag{4.5}$$

Then

$$P(z) = A(z) + z^{-(p+1)} * A(z^{-1}) \tag{4.6}$$

$$Q(z) = A(z) - z^{-(p+1)} * A(z^{-1}) \tag{4.7}$$

where $P(z)$ and $Q(z)$ are (p+1)-order symmetric and antisymmetric polynomials whose zeros are mapped onto the unit circle in the Z-domain. The zeros of the polynomial P and the zeros of the polynomial Q are interlaced with one another. The frequencies at which these zeros occur are the LSP frequencies. The zeros of the P polynomial are computed using the discrete cosine transform (DCT), whereas the zeros of the Q polynomial are computed using the discrete sine transform (DST).

The DCT is defined as follows:

$$f_j = \sum_{k=0}^{n-1} x_k \cos(\frac{\pi}{n} j(k + \frac{1}{2})) \tag{4.8}$$

and the DST is defined as:

$$f_j = \sum_{k=0}^{n-1} x_k \sin(\frac{\pi}{n} (j + 1)(k + \frac{1}{2})) \tag{4.9}$$

The DCT and DST are each similar to the DFT, but they both use real numbers and are roughly twice the length of the DFT. The DCT operates on data with even symmetry and is equivalent to the real parts of the DFT. The DST operates on data with odd symmetry and is equivalent to the imaginary parts of the DFT.

LSP frequencies have traditionally been used for speech data compression and quantization. One example of where they are used is in the well-known CELP cell phone data compression (Campbell, Tremain, and Welch, 1991), (Campbell, Tremain, and Welch, 1990). The LSP frequencies have been shown to be more stable than their LPC representation, therefore yielding higher accuracies in recognition applications. According to (Soong and Juang, 1984), LPC coefficients are inappropriate for quantization for several reason. They have a large dynamic range and are prone to filter instability problems, whereas LSP parameters have a well-behaved dynamic range and the property of filter stability.

Figure 4.2 (Zheng, Song, Li, Yu, and Wu, 1998) illustrates the relationship between the zeros of the P and Q polynomials and the corresponding transfer function $H(z)$. The closer together two adjacent zeros, the more of a peak is seen in the transfer function. (Zheng et al., 1998) exploits this relationship between the zeros of the polynomials and the transfer function $H(z)$ to derive a new set of distance metrics for LSP features. However, this technique is not within the scope of this thesis.

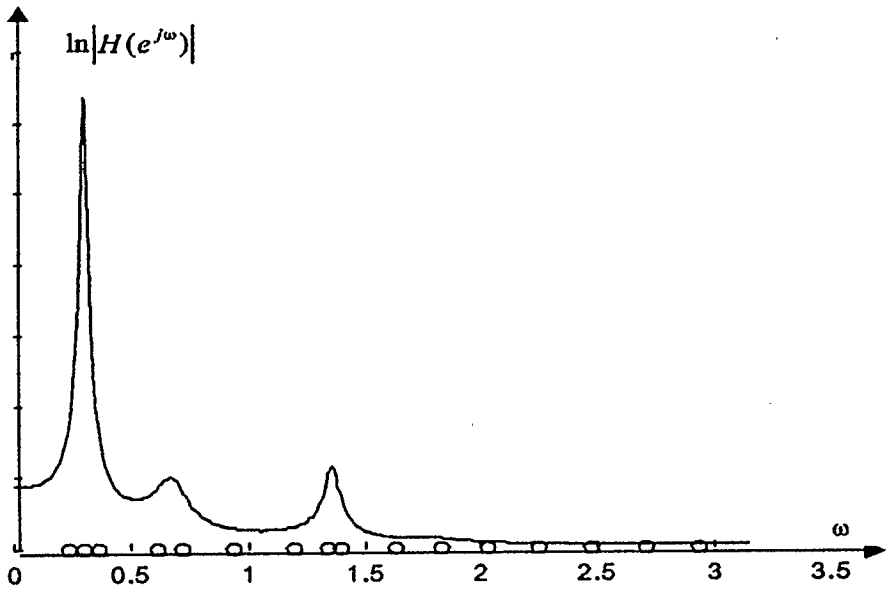


Figure 4.2: Zeros of P and Q polynomials and their relationship to the all-pole transfer function H(z). (Zheng, Song, Li, Yu, and Wu, 1998)

It has been shown (http://mi.eng.cam.ac.uk/~ajr/SpeechAnalysis/node1.html) that when a root of such a polynomial is close enough to the unit circle, it represents a formant. Formants, which are perceptually defined, correspond to the physical property of the frequencies of the resonances of the vocal tract. Formants are distinguishing components of human speech. They are the characteristic harmonics that identify vowels to the listener and allow the listener to distinguish between vowels (http://encyclopedia.thefreedictionary.com/Formant). Since the roots of $P(z)$ and $Q(z)$ lie directly on the unit circle, they represent formants. The angle at which a root lies is close to the fundamental frequency of the corresponding formant.

This has been explored in (Ribeiro and Trancoso, 1996). More specifically, $P(z)$ represents the portions of the acoustic tube model where the glottis is closed, and $Q(z)$ represents the portions of the model where the glottis is opened. The interleaving of the roots of $P(z)$ with the roots of $Q(z)$ around the unit circle is done to make the vocal tract model stable.

As stated in (Ribeiro and Trancoso, 1996), "the closer two consecutive LSP coefficients are together, the narrower the bandwidth of the corresponding pole of the vocal tract filter" (p. 307), and the higher the spectral peak. The further apart they are, the flatter the spectral peak. So, while "formants are marked by two close LSP coefficients, spectral tilt, or the slope of the spectrum, is primarily marked by LSP coefficients which are farther apart." (p. 307). This can be seen in figure 4.2.

According to (Ribeiro and Trancoso, 1996), "the roots of $P(z)$ have been named as position coefficients, because the closed glottis model is the best approximation for a lossless approximation of the vocal tract filter. Hence, whenever formants are present, one can find a correspondance between the roots of $P(z)$ and the locations of the formant frequencies. The roots of $Q(z)$, on the other hand, have been called difference coefficients, because of their role in marking the presence and absence of a formant by their closeness to a position ($P(z)$) coefficient." (pg. 307).

4.2.3 Psychoacoustic Features

Psychoacoustics is a science where data is gathered qualitatively on the perceptual auditory experiences of humans in order to try and gain a better understanding of how the human auditory system processes complex sounds. This data can be used to build synthetic models of the human auditory system, but direct physiological observation is limited to the outer, middle, and inner ears. Processing that occurs at higher neural centers cannot yet be studied in any great detail.

4.2.3.1 The Relationship Between Frequency and Position

The basilar membrane in particular has been the topic of much interest and research in the field of speech analysis. The ear's ability to analyze complex acoustic signals and target specific sounds in the presence of noise has led researchers to try and map out the inner workings of the ear. Figure 4.3 (Goldstein, 1999) shows a cross-section of the inner ear, in which the basilar membrane can be seen.

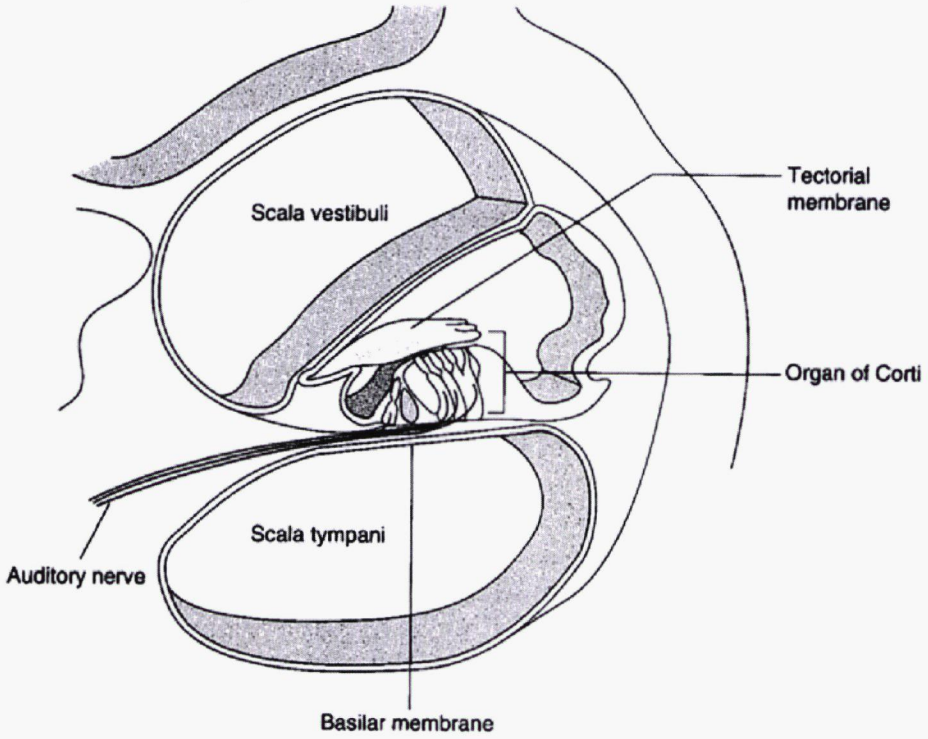


Figure 4.3: Inner ear cross-section. (Goldstein, 1999)

The combined movements of the basilar membrane and the tectorial membrane, which sits on top of the basilar membrane, in addition to the motion of the fluid within the inner ear, generates an electrical signal that is transmitted to the auditory nerve fibers and on up into higher neural centers of the brain.

4.2.3.2 Hermann von Helmholtz (Resonance Theory)

Hermann von Helmholtz was the first scientist to propose the idea that certain parts of the ear are responsible for dealing with specific frequencies (Morgan and Gold, 2000), (Goldstein, 1999). Using the knowledge that the basilar membrane is narrow at one end and wider at the other, Helmholtz hypothesized that the basilar membrane is made up of a

series of adjacent fiber strips (much like a xylophone) that start narrow at the base and get wider as they move toward the apex. He proposed that each fiber strip was tuned to a specific frequency, such that high frequencies would set long fibers into vibration and low frequencies would set short fibers into vibration, much like the strings in a piano, where each fiber is able to resonate independent of the others. The cilia on a stimulated fiber strip would then be bent, causing only the transduction of these vibrations into electrical signals to be passed upward.

Helmholtz' theory has been found to be incorrect. It was later found that large regions of the basilar membrane react to sound stimulus because the basilar membrane's fibers are connected and thus cannot act independently of other fibers. However, his work inspired others, such as George Bekesy, to use his ideas as a springboard for other work.

4.2.3.3 George von Bekesy (Traveling Wave Theory)

Bekesy (Morgan and Gold, 2000), (Goldstein, 1999), (Bekesy, 1942) performed an experiment on the basilar membrane to find out how it behaves when stimulated by vibrations of different frequencies. He explored the behavior of the basilar membrane not only by directly observing the reaction of the basilar membrane itself, but also by constructing a model of the cochlea. The cochlea was constructed to support the fact that the basilar membrane is narrower and more stiff at one end and wider and less stiff at the other.

Figure 4.4 (Hermansky, 1990) depicts a filterbank of synthetic filters with a trapezoidal shape. These are what are used by Hynek Hermansky in his PLP analysis, which is discussed in section 4.2.3.8. Although these filterbank shapes were derived empirically from testing the limits of the ear, they are not necessarily those applied by the physical ear. Several researchers have proposed various other filter shapes to model what the ear is doing. As shown in the figure, several bandpass filters cover the perceptible region of the frequency-domain spectrum. Thus, a given range of frequencies is filtered through any filters that cover that region, and are analyzed separately from other frequencies that do not lie within this bandpass region.

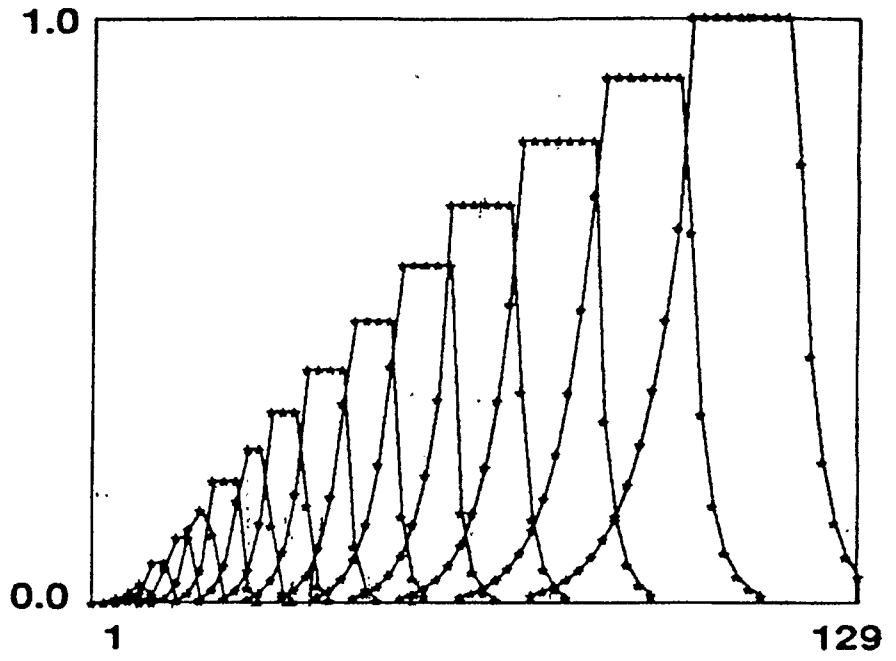


Figure 4.4: A filterbank of trapezoidally-shaped filters. (Hermansky, 1990)

This would mean that the ear is able to break down a complex signal into its different frequency components, much like the Fourier transform. Many researchers have performed experiments to explore the nature of these auditory filters.

4.2.3.5 Masking

Suppose a tone is introduced to a person, and a second tone is introduced later on. If the second tone is intense enough, it can mask, or decrease, the person's perception of the first tone. This phenomenon in human auditory perception is called masking. A tone can more readily mask a tone that it is closer to in frequency than one that is further away. In addition, a tone can more readily mask a tone that is at a higher frequency than itself, rather than a tone that is at a lower frequency (Goldstein, 1999).

4.2.3.5.1 Harvey Fletcher (Critical Bands)

Harvey Fletcher (Morgan and Gold, 2000) performed a series of experiments to further test the frequency-specific positioning characteristics of the basilar membrane by trying to find the bandwidths and spectral positions of the auditory filters in the proposed filterbank. To do this he performed an experiment that is referred to as simultaneous masking, which exploits the masking phenomenon that was described above.

In Fletcher's simultaneous masking, a test tone plus wideband noise is presented to a listener. The wideband noise is centered at the frequency of the test tone. The test tone is decreased in intensity until it can just barely be perceived, and decreasing the intensity any further would make it so that the tone could no longer be perceived. The intensity at

which the tone is no longer perceived is referred to as the threshold intensity. It is the lowest intensity needed for the person to be able to hear the tone in the presence of noise. The noise bandwidth is then decreased, causing a decrease in the power of the noise, and the intensity threshold of the test tone is recomputed. No change occurs in the intensity threshold of the test tone until ae "critical band" is reached. As the noise bandwidth is decreased further and further from this critical band, the intensity threshold decreases as well.

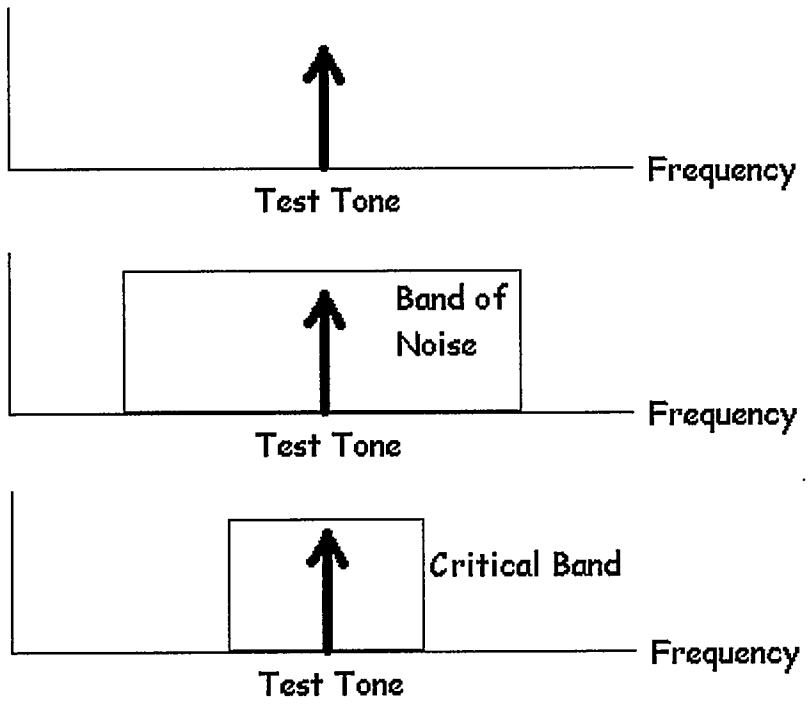


Figure 4.5: Narrowing noise bandwidth to find the critical band.

The idea is that a small bandwidth of noise will contribute a small amount of masking to the test tone, and the intensity threshold will be smaller in the presence of this noise (i.e. -

it is easier to hear the tone than if there is a wider bandwidth of noise). As the bandwidth of the noise increases, more masking is applied to the test tone, increasing the amount of intensity needed for the person to perceive the tone. This increase in intensity threshold eventually levels off beyond the critical band, since noise outside of the critical bandwidth no longer has any effect on the test tone, supporting the idea of a filterbank of bandpass auditory filters. When a person is trying to perceive a tone in the presence of noise, the auditory filter with a central frequency that is closest to the frequency of the test tone is used to filter out all noise outside of the filter and only pass the noise and test tone within the filter. Thus it is only this noise that has any masking effect on the tone.

It has been found that the bandwidths of the auditory filters increase with increasing central frequency. It has also been found that the auditory filters increase in size with increasing central frequency.

These experimental findings were later confirmed by other researchers, such as Egan and Hawke (Egan and Hake, 1950). Although these experiments have yielded information about the bandwidths and frequency positions of the filters, they have unveiled nothing about the shapes of the filters themselves. Experiments performed to approximate the shapes of the auditory filters are briefly described in section 4.2.3.7.

4.2.3.6 Natural Frequency Scale of the Ear (Bark Scale)

In (Schroeder, 1977), Schroeder explains that the frequency resolution of the ear is around 1.5mm, and that the total length of the basilar membrane from the base to the apex is roughly 35mm. This gives around 24 critical bands and thus 24 auditory filters.

Schroeder explains that the ear has its own natural frequency scale, which corresponds to equal distances along the basilar membrane. This is referred to as the Bark scale. A frequency f is warped into a Bark z by:

$$f = 600 * \sinh(\frac{z}{6}) \tag{4.10}$$

Thus, the human perception of sounds at different frequencies is not linear. This has implications for critical bands. Bandwidths of critical bands with a central frequency below 500 Hz are constant. Above 500 Hz, the bandwidths increase in a roughly logarithmic fashion as functions of the central frequency of the bandwidth. This follows Weber's law, which states that our peripheral senses tend to follow a logarithmic law of sensation in response to a stimulus.

4.2.3.7 Auditory Filter Shapes

Over the years, theoretical auditory filter shapes, as well as the methods that have derived them, have become increasingly more complex.

First, a set of psychoacoustic measurements are obtained. A tone, whose frequency is varied between 400 to 1400 Hz, is presented to a listener. A rectangular high-pass band of noise is present from 1200 Hz to 1400 Hz, and a rectangular low-pass band of noise is present from 400 Hz to 600 Hz. The intensity threshold of the listener's perception of the tone is computed at each frequency of the tone.

A derived auditory filter shape must match these psychoacoustic intensity thresholds. Several filter shapes have been proposed that approximate the results. Fletcher first proposed an ideal rectangular filter shape, although he knew this was an erroneous assumption. The results obtained from a rectangular filter shape deviate the most from the psychoacoustic results. He chose to use this filter anyway because the particular shape of the filter did not matter in the calculations he was performing (Morgan and Gold, 2000). The problem with this method of obtaining a filter shape is that, while the fixed, absolute filter shapes could be predicted, arbitrary filter transfer functions could not be designed directly from the psychoacoustic measurements.

Roy Patterson (Patterson, 1976) did some important work on deriving auditory filter shapes that could have this arbitrary filter transfer function. He began by varying the width of the rectangular noise bands, while keeping the tone frequency fixed. For each change in the bandwidth of the noise, the intensity at which the tone was just barely heard was computed. From this, he was able to build a mathematical representation of the relationship between the noise and the predicted filter, and was able to compute a magnitude function of the transfer function of the filter, with the assumption that the

auditory filters were centered around the frequency of the test tone. He was unable to derive the phase of the filter from this. He also found that the band of noise interfered too much with the detection of the tone when the noise bandwidth got too close to the frequency of the tone. In addition, he realized that the assumption that human auditory filters were centered around the tone to be detected was incorrect, meaning that the central frequency around which the filter is positioned is not always equal to the frequency of the test tone.

Patterson accounted for these problems by using notched wideband noise, in which listening that was off-frequency caused the noise to shift so that the total masking of the noise stayed the same. From this, Patterson constructed a symmetric filter which has parameters that can be selected arbitrarily, which accurately approximates the psychoacoustic measurements.

$$|H(f)|^2 = \frac{1}{\left[(\Delta f/\alpha)^2 + 1\right]^2} \tag{4.11}$$

Other filter derivation methods that have been explored include Gamma-tone filters and Roex filters (Morgan and Gold, 2000). These are not discussed here.

4.2.3.8 Perceptual Linear Prediction (PLP) Coefficients

Hermansky's Perceptual Linear Predictive analysis of speech exploits many of the aforementioned psychoacoustic properties of human hearing and is a popular feature that

had been used in many ASR systems (Hermansky, 1990). The procedure for analyzing a frame of speech to yield a set of PLP coefficients is described briefly next.

Hermansky begins by applying a hamming window to the time-domain speech signal within the frame to prepare it for frequency-domain analysis. The windowed signal is taken to the frequency domain using Fourier analysis, and the power spectrum is computed:

$$P(\omega) = \mathrm{Re}al[S(\omega)]^2 + \mathrm{Im}ag[S(\omega)]^2 \qquad (4.12)$$

Next, a filterbank of auditory filters are constructed, simulating what has been found about the basilar membrane's response to sound stimuli. Each auditory filter is quasi-trapezoidal in shape. The full filterbank is shown in figure 4.6.

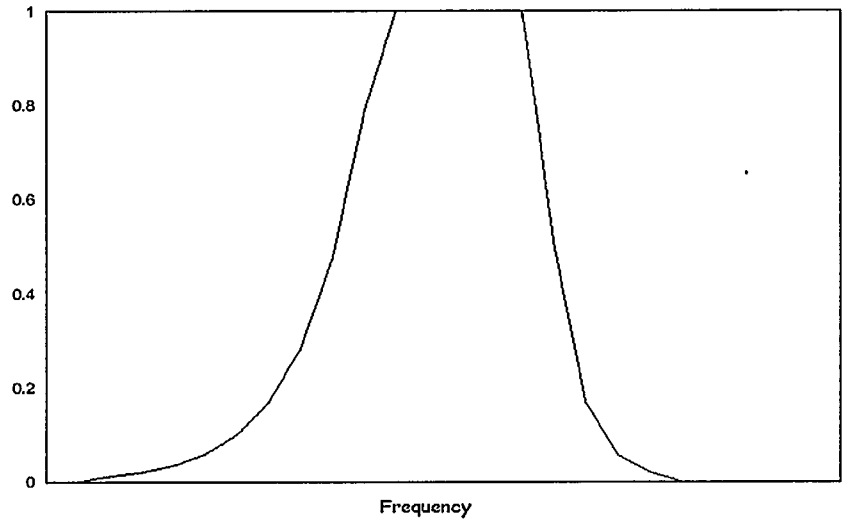


Figure 4.6: The shape of Hermansky's auditory filters.

First, the critical bands are computed in Bark frequencies, and then the filters themselves are built around them. Given the central frequency of a critical band, the auditory filter is built using:

$$\psi(\Omega) = \begin{cases} 0 & for\ \Omega < -1.3 \\ 10^{2.5(\Omega+0.5)} & for\ -1.3 \le \Omega \le -0.5 \\ 1 & for\ -0.5 \le \Omega \le 0.5 \\ 10^{-1.0(\Omega-0.5)} & for\ 0.5 \le \Omega \le 2.5 \\ 0 & for\ \Omega \ge 2.5 \end{cases} \tag{4.13}$$

The frequency boundaries of each of the critical bands of the filters are warped using Schroeder's Bark (z) to Frequency (f) equation:

$$f = 600 \sinh\left(\frac{z}{6}\right) \tag{4.14}$$

This is equivalent to warping the power spectrum $P(\omega)$ along its frequency axis into Bark frequencies, and then applying the auditory filters directly, rather than inverse warping them.

Next, Hermansky approximates the nonequal sensitivity of human hearing at different frequencies using equation 4.15:

$$E(\omega) = \frac{[(\omega^2 + 5.68*10^6)\omega^4]}{[(\omega^2 + 6.3*10^6)^2(\omega^2 + 0.38*10^9)(\omega^6 + 9.58*10^{26})]} \quad (4.15)$$

In human hearing, the stimulus at one frequency will be perceived as louder than another stimulus of the same intensity that is at a different frequency. The resulting auditory filterbank is then convolved with the power spectrum $P(\omega)$.

$$\Xi(\Omega(\omega)) = E(\omega)\Theta(\Omega(\omega)) \quad (4.16)$$

An approximation to the power law of hearing is next applied, which is computed by:

$$\Phi(\Omega) = \Xi(\Omega)^{0.33} \quad (4.17)$$

Finally, the pre-processed frequency spectrum $\Phi(\Omega)$ is transformed back to the time-domain, and linear prediction coefficients are extracted to yield an all-pole autoregressive model of the speech. This gives us the PLP coefficients. This autoregressive property of the PLP features is similar to that of the LPC features, a property that is exploited to create the experimental PLP-LSP features in chapter 5.

Perceptual Linear Prediction coefficients are used quite often for automatic speech recognition. Speaker-dependent information is largely stripped away, leaving the linguistic content, given that the autoregressive model is of a low order. Hermansky has

reported that PLP coefficients of order 5 yields these characteristics best. Higher orders contain speaker-specific as well as lexical content.

4.3 Classifiers

In figure 4.7 (http://www.haifa.il.ibm.com/Workshops/Speech2003/papers/IBM_03.pdf), a timeline is shown that depicts the shifting trends in the choice of classifiers for speaker recognition. As shown, there has been a shift from template classifiers to statistical classifiers in response to a shift from smaller databases recorded under more pristine laboratory conditions to larger databases recorded under more realistic, spontaneous, and noisy conditions. Some of the more commonly used classifiers will be briefly described in this section.

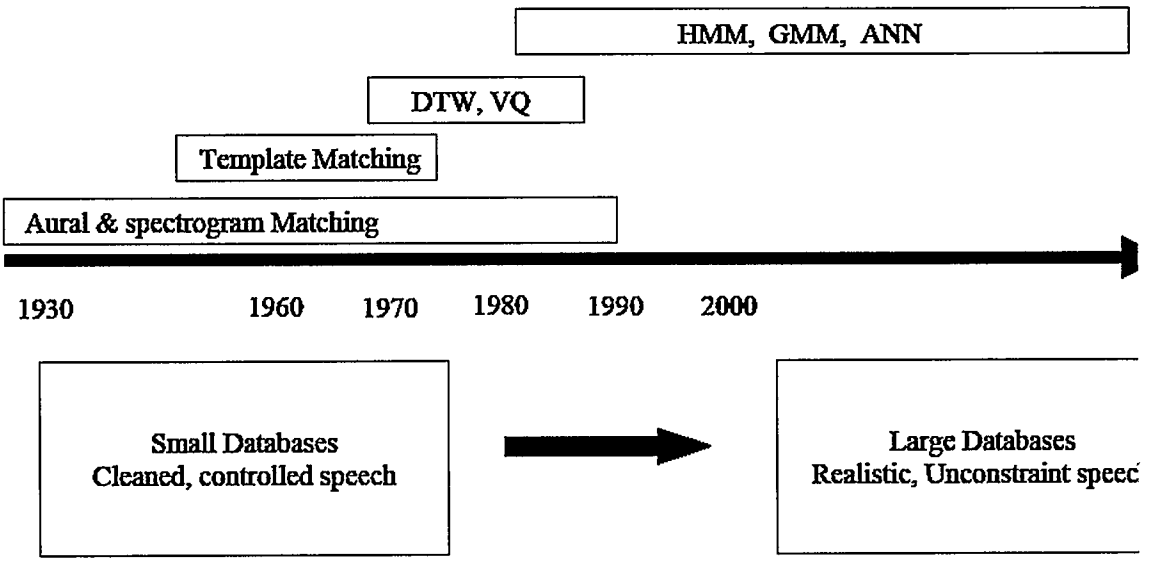


Figure 4.7: Classifier timeline
(http://www.haifa.il.ibm.com/Workshops/Speech2003/papers/IBM_03.pdf)

This general shift from template to statistical classifiers has resulted in marked improvements in speech and speaker recognition in recent years. Not only are statistical models more robust under noise, they are more flexible in allowing for slight variations that were not seen in the training speech data. As already discussed, for speech recognition applications especially, there are usually too many variations to be able to adequately capture within the training set. It would take enormous amounts of input data in order to do a half-decent job, and the computational expense is unrealistic.

4.3.1 Template Classifiers

Template classifiers model a given class using what is called a template. The template is considered to be the most representative instance of all observations belonging to that class. Therefore, when classifying a new unknown observation, a distance measurement is performed between the new observation and the template observation.

For exemplar-based systems, there are a few variations of this idea. An exemplar is one of several observations belonging to a class. The first, called "nearest neighbor", is where a new observation is compared to all observations of all the classes. The observation is said to belong to the class containing the observation that is nearest to the new observation. The second method, which is called k-nearest neighbor, is a variation of the first. It is the same idea, only this time the observation is said to belong to the class containing the k closest observations to the new observation. Finally, the centroid method is where the centroid of all the observations is computed and is used as the representative observation. The new observation is then compared with the centroids of all the classes to see which one it is closest to. One issue with this method is that the centroid is usually not actually seen in the training data, and therefore is somewhat synthetic. However, if it lies within the boundaries of the physical space within which the class lies, then it probably doesn't make much difference.

4.3.1.1 Dynamic Time Warping (DTW)

Speech is a time-dependent process. Several utterances of the same word are likely to have different durations, since they are never spoken in the same way or at the same rate.

Most algorithms that process time series data need to compute some sort of similarity between the time series. Euclidean distance or some extension is commonly used. However, Euclidean distance can be an extremely "brittle" distance measure. The reason why Euclidean distance may fail to produce an intuitively correct measure of similarity between two sequences is because it is very sensitive to small distortions in the time axis. Comparing two utterances of the same word in time using a distance measure will not work since they do not correspond at the same points in time. Consider the left part of figure 4.8 (Keogh and Pazzani, 2000):

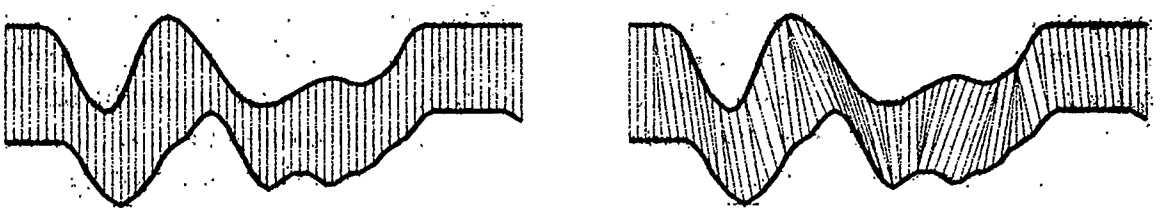


Figure 4.8: (Left) Linear alignment of two sequences. (Right) Time-warped alignment of the two sequences. (Keogh and Pazzani, 2000)

In the figure, two sequences are being compared, one on the top and one on the bottom. The two sequences have approximately the same overall shape but those shapes are not exactly aligned in the time axis. A time alignment must first be performed in order to obtain a global distance between two speech patterns (represented as a sequence of vectors). The non-linear alignment in part B would allow a more sophisticated distance measure to be calculated.

Dynamic Time Warping (Keogh and Pazzani, 2000), (http://www.dcs.shef.ac.uk/~stu/com326/sym.html) is a method for achieving such alignments that has been used for a long time in speech processing. However, the algorithm is very computationally expensive, and usually takes too long to be of any practical use in a real-world application.

The idea behind the algorithm is shown in figure 4.9 (http://www.dcs.shef.ac.uk/~stu/com326/sym.html).

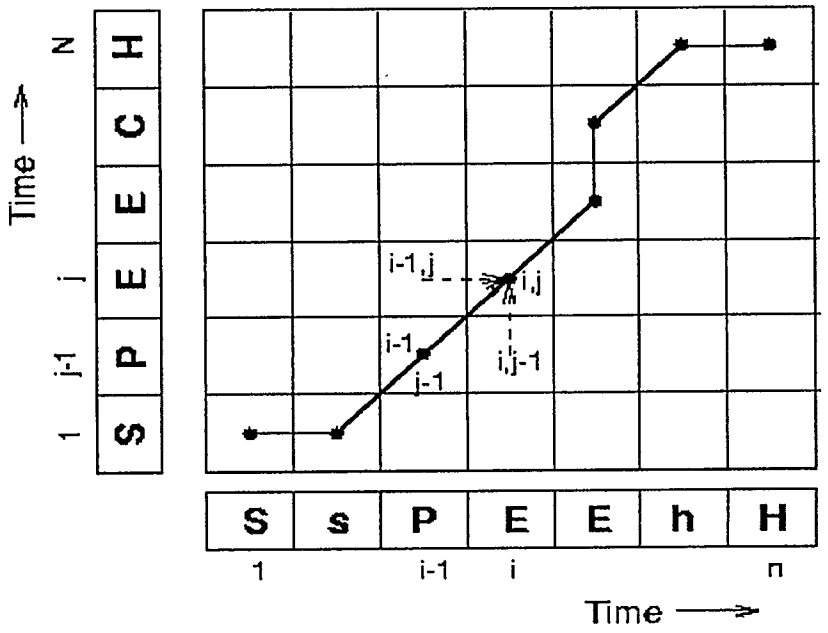


Figure 4.9: Dynamic time-warping alignment of two different utterances of the word "Speech". (http://www.dcs.shef.ac.uk/~stu/com326/sym.html)

In figure 4.9, two different utterances of the pattern "Speech" are being time-aligned. There are two axes: the vertical axis represents the first form of the pattern, and the

horizontal axis represents the second. Each frame (or feature vector) of the vertical pattern creates a row in the 2D matrix, while each frame of the horizontal pattern creates a column. A path must be found through the matrix, starting at the lower left index and ending at the upper right index, such that the distance between the two patterns at each 2D index is minimized. To find the best matching path, and ultimately the best global distance between the two patterns, it is highly inefficient to consider all possible paths through the matrix. The best matching path between the two can be discovered much more efficiently by imposing a few simple rules:

- matching paths cannot go backwards in time
- every frame in the input must be used in a matching path
- local distance scores are combined by adding to give a global distance

Clearly the path through the matrix is not linear, and thus illustrates how the time axis must be warped to align the two patterns in time.

4.3.1.2 Vector Quantization (VQ)

Vector Quantization (VQ) (Pop and Lupu, 2002), (http://www.data-compression.com/vq.html) is the process of taking a large set of feature vectors and producing a smaller set of feature vectors that represent the centroids of the distribution, i.e. points spaced so as to minimize the average distance to every other point.

We begin by visualizing a set of feature vectors in 2-dimensional space. These are represented by the x's in figure 4.10 (Pop and Lupu, 2002). A codebook is built around these feature vectors. It consists of a set of Voronoi regions. For each region there is a codeword, or codebook vector, that acts as a representative vector for all the feature vectors that lie within that particular Voronoi region. This is known as quantization. All feature vectors are mapped onto one of the codebook vectors.

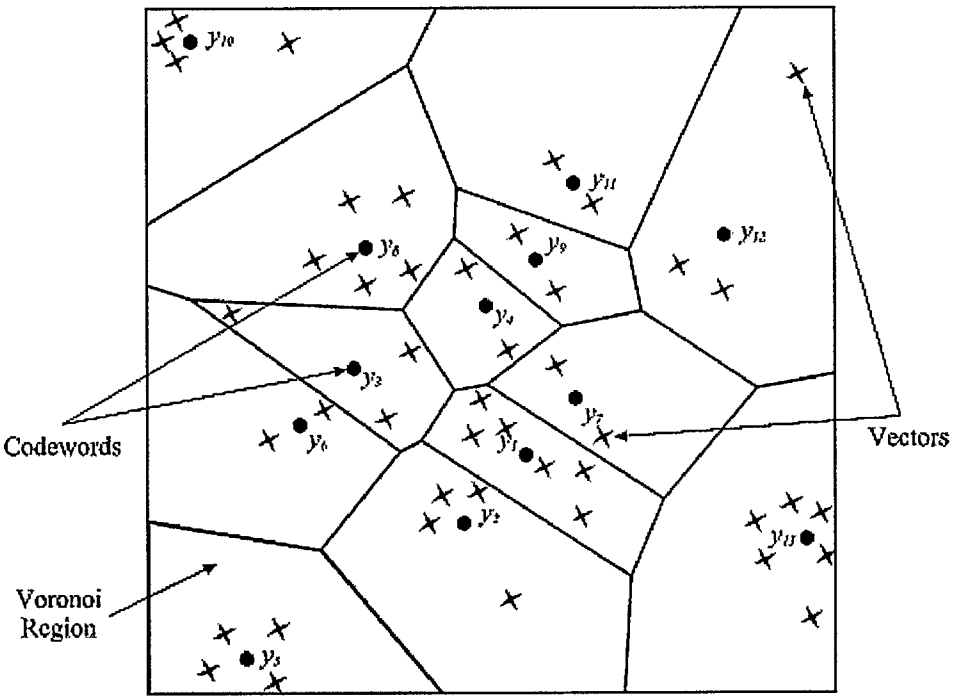


Figure 4.10: A 2-dimensional codebook. Dots are codewords, x's are feature vectors, and the shaded area is a Voronoi Region. (Pop and Lupu, 2002)

The most common and simplest codebook training algorithm is the Linde-Buzo-Grey (LBG) algorithm given in (http://www.data-compression.com/vq.html), so named for the authors who created it. In the algorithm, a codebook is trained by first starting with one big Voronoi region and one single codebook vector that is at the center of all the feature

vectors. A distortion measure (usually a Euclidean distance or squared-error distance) is computed from the codebook vector to each of the feature vectors within the Voronoi region. The average distortion is the mean of all the distortions over all the Voronoi regions that exist at a given time. The Voronoi region is then split in half and two new codebooks are chosen for the two new Voronoi regions such that the average distortion between the codebook vectors and the feature vectors that lie in the Voronoi regions is minimized. This region splitting and refitting of codebook vectors is continued until the desired number of Vornoi regions is reached. This results in a codebook similar to the one shown in figure 4.10. The codebook represents the best fit to the training vectors that were supplied for that person. The larger the number of regions, the better the speaker model representation.

Although there are several methods that can be used to construct a codebook for a speaker, such as self-organizing maps or simulated annealing, it has been found that there is only a marginal difference in accuracy between the codebooks constructed using each of the methods. The LBG algorithm, which is described in (http://www.data-compression.com/vq.html), is commonly used in the literature since it is the simplest.

In the testing phase, an input test utterance is obtained from the unknown user and the feature vectors are extracted in the same way as those that were used to generate the codebook. These feature vectors are then encoded using the target speaker's codebook. A distortion measure is computed between each input feature vector and the codebook vector that is representative of the corresponding Voronoi region in which the feature

vector lies. The average distortion over all the distortion measure between each feature vector and its associated codebook vector is obtained.

There are three different types of VQ that have been used in speech analysis: single-section (Pop and Lupu, 2002), multi-section (Pop and Lupu, 2002), and matrix (Burton, 1985) VQ. The details of these are not described here.

4.3.2 Statistical Classifiers

Statistical classifiers are better-suited to realistic speaking environments and have gained much popularity in the last couple of decades. It is well-known that Gaussian Mixture Models (GMM) are one of the most flexible classifiers for text-independent speaker recognition and therefore they are able to achieve higher recognition performance in a more realistic setting. This is because they are more able to model variations in the speaker's voice samples than other classifiers, and are also more able to take into account noisy environments. Other statistical classifiers, such as pdf-based methods, only make use of a single pdf component, whereas the GMM makes used of several, thereby allowing it to more tightly model a broader range of voice characteristics. Hidden Markov models (HMM) (Rabiner and Juang, 1986), (Yun and Oh, 2000), (Rabiner, 1989) and HMM hybrids (Nakamura and Markov, 2004), (Boite and Ris, 1999) model temporal information, and are therefore better suited to text-dependent speaker recognition, in which pattern-matching the lexical content of an utterance is more important. For text-independent tasks, GMM's are a better choice than HMM's. HMM's are not in the scope of this thesis and are not discussed.

4.3.2.1 Probability Density Function

The normal Probability Density Function (PDF) (Stark and Woods, 2002) is a good approximation to real-world density functions. It is one of the simplest parametric models, being characterized by a mean and variance. There are several methods of comparing one PDF to another directly. These include the Mahalanobis distance, the Divergence measure, and the Bhattacharyya distance, each of which is described briefly in this section.

An n-variate normal PDF is defined as:

$$p(x) = (2\pi)^{\frac{n}{2}} |C|^{\frac{1}{2}} \exp\left[-\frac{1}{2}(x-\mu)^T C^{-1}(x-\mu)\right] \tag{4.18}$$

Classifers that make use of such distortion comparisons are represented by a covariance matrix and a mean vector as a PDF.

4.3.2.1.1 Mahalanobis Distance

Ignoring the factor of $-1/2$, the argument of the exponent in equation 4.18 is referred to as the Mahalanobis distance (http://www.unesco.org/webworld/idams/advguide/Chapt9_3.htm) between x and μ. According to (Campbell, 1997), "the loci of points of constant density are hyperellipsoids of constant Mahalanobis distance to μ. Samples drawn from a multivariate normal

density tend to cluster. The center of the cluster is determined by the mean vector, and the shape of the cluster is determined by the covariance matrix". As shown in figure 4.11 (Campbell, 1997), groups of samples that exhibit high variance are more stretched and elliptical in shape, while groups of samples that exhibit low variance are more circular in shape.

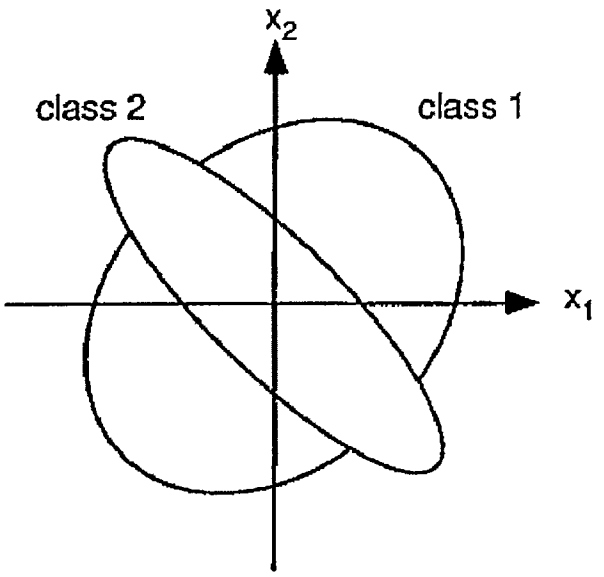


Figure 4.11: Ellipsoidal clustering of two different classes based on Mahalanobis distance. (Campbell, 1997)

4.3.2.1.2 Divergence Measure

Divergence (Kailath, 1967) is a measure of distance or dissimilarity between two classes based upon information theory and is defined as follows:

$$J_{ij} = tr[(C_i - C_j)(C_j^{-1} - C_i^{-1})] + \frac{1}{2} tr[(C_i^{-1} - C_j^{-1})\delta\delta^T] \qquad (4.19)$$

where C_i and C_j are covariance matrices, $\delta = \mu_i - \mu_j$ is the difference in means, and J_{ij} is the Divergence. The Divergence is a sum of a size component and a shape component, respectively. More specifically, it is a sum of the average shape and difference in size of the two PDF's being compared.

The Divergence Shape component is:

$$J_{ij}' = tr[(C_i - C_j)(C_j^{-1} - C_i^{-1})] \tag{4.20}$$

while the remainder of equation 4.19 corresponds to the size. In some cases, only the shape component is used in the distortion measure.

4.3.2.1.3 Bhattacharyya Distance

As with the Mahalanobis distance and the Divergence measure, the Bhattacharyya distance (Kailath, 1967) directly compares the estimated mean vector and covariance matrix of a test segment with those of the target voice model. It is defined as:

$$d_B^2 = \frac{1}{2}\ln\left(\frac{\left|\frac{C_i + C_j}{2}\right|}{|C_i|^{\frac{1}{2}}|C_j|^{\frac{1}{2}}}\right) + \frac{1}{8}(\mu_i - \mu_j)^T\left(\frac{C_i + C_j}{2}\right)^{-1}(\mu_i - \mu_j) \tag{4.21}$$

As with the Divergence measure, the Bhattacharyya distance is the sum of a shape component and a size component, respectively.

4.3.2.2 Gaussian Mixture Model (GMM)

GMM classifiers (Reynolds, Quatieri, and Dunn, 2000), (Xiang, Chaudhari, Navratil, Ramaswamy, and Gopinath, 2002) use probabilistic measurements to determine class membership, or how "likely" it was that the class generated the new observation. The Gaussian probability distribution function (pdf) is a common statistical classifier used for speech and speaker recognition. Given a set of data, it is possible to estimate the parameters for a Gaussian pdf that best fits the data.

A single Gaussian pdf describing a k-dimensional random variable X has the form:

$$g(x) = \frac{1}{(2\pi)^{\frac{k}{2}}|C|^{\frac{1}{2}}} \exp\left(-\frac{1}{2}(x-\mu)^T C^{-1}(x-\mu)\right) \tag{4.22}$$

where C is the covariance matrix, |C| is its determinant, and μ is the mean vector.

A more sophisticated extension of the Gaussian pdf is the Gauss mixture pdf. A Gauss mixture pdf is a weighted sum of a collection of distinct Gaussian pdf's.

An n-variate Gaussian density is defined as:

$$b_i(x) = \frac{1}{(2\pi)^{\frac{k}{2}}|C_i|^{\frac{1}{2}}} \exp\left(-\frac{1}{2}(x-\mu_i)^T C_i^{-1}(x-\mu_i)\right) \tag{4.23}$$

$$p(x|\lambda) = \sum_{i=1}^{M} p_i b_i(x) \tag{4.24}$$

where

$$\sum_{i=1}^{M} p_i = 1 \tag{4.25}$$

and where $b_i(x)$, $i = 1,...,M$ are the component densities and p_i is the weight of the i^{th} mixture component.

When used as a classifier, this is referred to as a Gaussian Mixture Model (GMM), where each training observation is interpreted as having been generated by one of the Gaussians. The Gaussian mixtures are built around the input training data in a best-fit manner. This is accomplished using the Expectation-Maximisation (EM) algorithm (Moon, 1996) in an iterative fashion to estimate the parameters of each Gaussian.

GMM's are used in both speech and speaker recognition. In speaker recognition, they are commonly used in text-independent applications because of their flexible nature and their

ability to take into account slight variations in the data, as well as noise. In speech recognition they are commonly used in conjunction with HMM's.

4.3.2.3 Neural Networks

Neural networks (Lawrence, Burns, Back, Tsoi, and Giles, 1998), (Kasuriya, Wutiwiwatchai, Achariyakulporn, and Tanprasert, 2001) are also frequently used in speech analysis. However they are often not used on their own. Used in combination with statistical methods, such as HMMs, they make a very powerful hybrid classifier. These, as well as GMM-HMM hybrids (Rodríguez, E., Ruíz, B., García-Crespo, Á., and García, F., 1997), (Huang, Chen, and Chang, 2002), (Vandecatseye and Martens, 2003) have yielded enormous gains for speech recognition applications in particular. Such hybrids are generally used for text-dependent recognition, however, because of the ability of the HMM to model temporal information, such as lexical speech content.

A neural network is made up of a set of nodes that are interconnected. These nodes are primitive models of neurons in the brain. The inputs to a node are processed and produce an output. The output can be fed as input to another set of nodes. This output is referred to as the "activation" of the neuron, and is analogous to the firing response of a neuron. Figure 4.12 depicts a typical node.

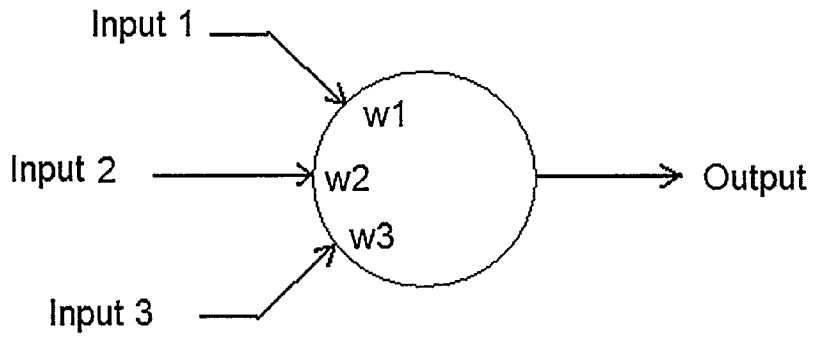


Figure 4.12: A simple node, or neuron, in a neural net.

The node's activation output is computed from its activation function as follows:

$$activation = \sum Input_i * w_i \tag{4.26}$$

The activation function can be made to ensure that the activation output falls within a legal range. For example, a node can be defined to accept binary inputs and deliver a binary output. This can be accomplished through normalization or some sort of thresholding.

A neural network contains several layers of nodes. The first layer is called the input layer. This layer accepts input from input training feature vectors to be introduced into the network. The point of a neural net is to associate these inputs with some output. The training of the neural net updates the inputs and outputs of each of the nodes such that all training inputs generate their desired outputs. The last layer is the output layer, which generates the desired output. There can be 0 or more hidden layers in between the input

and output layers. The layers are connected as shown in figure 4.13. The 1st layer feeds the 2nd layer and so on up until the last layer.

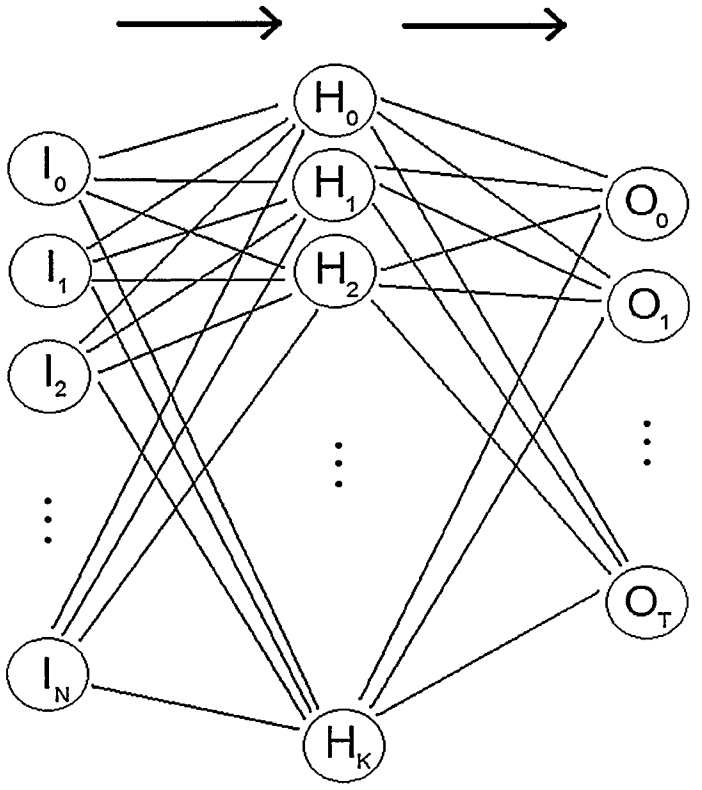


Figure 4.13: A connected neural network, with one input layer, one hidden layer, and one output layer.

A common method of training a neural net is called backpropagation. In backpropagation (http://www-gpi.physik.uni-karlsruhe.de/pub/robert/Diplom/node8.html), the input is presented to the input layer and propagates through each of the layers to the output layer, at which point the weights of the nodes of each layer are updated in a backward fashion. This is repeated until a suitable error has been achieved.

CHAPTER FIVE: SPEAKER RECOGNITION USING PLP-LSP FEATURES

While LSP features have traditionally been used for speech data compression, they have recently been shown to be effective for speaker recognition applications. As already mentioned in chapter 4, they have been shown to be a better choice than LPC features (Soong and Juang, 1984) in terms of stability and robustness under noise. While LSP features are starting to be considered for recognition applications, there is still much ground that hasn't been covered. Can the LSP frequencies be calculated for features other than LPC features for speaker recognition purposes? Do they add enhanced properties to the underlying features? This chapter addresses such questions. In particular, LSP frequencies of PLP features (PLP-LSP) are the main focus. PLP features were chosen because of their psychoacoustic nature, and their autoregressive similarity to LPC features. In my opinion, features that more closely follow the human model, which obviously works very well, need to be further explored.

As already mentioned in chapter 1, it is desirable to simulate what the human body is doing because it is an already working model of the pattern matching we are trying to accomplish. PLP coefficients were chosen for this experiment for several reasons. First, they are psychoacoustic in nature and try to simulate what the human ear is doing. Second, they already yield quite a high success rate, outperforming LPC coefficients in many recognition tasks, both in the literature (Hermansky, 1990), (http://www.asel.udel.edu/icslp/cdrom/vol3/706/a706.pdf) and in my own experiments which are described in section 5.2. Third, they have an autoregressive property that is

similar to LPC coefficients. It is this autoregressive similarity to LSP that prompted me to try representing PLP coefficients in an alternative LSP representation.

Note that there are other psychoacoustic features having roughly the same set of computational steps as PLP but with minor differences, such as the shape of the auditory filters in the filterbank, that do seem to outperform PLP. However, the idea is to see if the alternative LSP representation can be computed for one set of features within this class, namely PLP, to improve its recognition performance. This would make it highly likely that similar benefits can be gained for other features within that class. So, PLP is a reasonable choice of feature for this purpose.

The experiment discussed in this chapter compares the recognition performance of LPC, LSP, PLP, and PLP-LSP features in order to answer two questions. First, do PLP-LSP features outperform PLP features? And second, how do PLP-LSP features compare to LSP features? Do they do better, worse, or about the same? LPC features are included simply to illustrate that PLP outperforms them. A more direct question might be, given that PLP features outperform LPC features, do PLP-LSP features similarly outperform LSP features?

In (Campbell, 1997), Joe Campbell explained many key concepts in speech signal processing for speaker recognition. He performed a set of speaker recognition experiments in which he explored the effectiveness of different combinations of features and classification schemes. The experiments were done in a loosely text-dependent

manner using the YOHO speech corpus, and yielded high results. His highest accuracies came from using 10th-order Line Spectrum Pair (LSP) Frequencies in conjunction with a Divergence Shape distance measurement.

5.1 Joe Campbell's Experiment

In (Campbell, 1997), several combinations of different features and classifiers were evaluated using various pattern-matching techniques in a semi-text-dependent identification experiment. Among the features were Log Area Ratios, Reflection Coefficients, and Line Spectrum Pair frequencies, all of which are discussed in some detail in the paper. Each is derived from the initial LPC coefficients and is considered an alternative representation of these features. The speaker recognition performance of these features was tested using two different types of classifiers: a pdf was used first with a Bhattacharyya distance metric and then with a Divergence distance metric, both of which are described in chapter 4. The best results came from using LSP frequencies with a Divergence Shape distance metric. The correct identification rate for the LSP/Divergence Shape combination was around 98.9%, which is quite a high rate of success.

I describe it as semi-text-dependent because, though the speaker models were trained using several utterances of the same phrase for each speaker, a test utterance to be classified was not explicitly matched on lexical content. This means that while attempts were made to identify an unknown person by finding a best match to those speaker models in a database of speakers, there was no check to ensure that the phrase being spoken was the same phrase that was trained on.

5.1.1 Speech Database

The YOHO speech corpus (Campbell and Reynolds, 1999), (http://wave.ldc.upenn.edu/Catalog/readme_files/yoho.readme.html) is a database of 138 speakers (108 males, 30 females) set in a semi-noisy office environment. The speakers have varying geographic origins and thus have varying accents and intonations. Quiet occasional background noise is present, such as the odd telephone ringing or a fan blowing. The database was constructed specifically for speaker recognition tasks. Its vocabulary consists of two-digit numbers spoken continuously in sets of three. An example of such a phrase is "twenty-two, sixty-four, thirty-six". The corpus is divided into training and testing sets for each speaker. The training set consists of four enrolment sessions per speaker, each containing 24 utterances. The testing set consists of 10 sessions per speaker, each containing four utterances. In (Campbell, 1997), only 44 of the 138 speakers were used for the speaker recognition task.

5.1.2 Overview

In this section, each of the steps in the classification pipeline, such as framing, frame windowing, feature extraction, and modeling and classification, will be described in the context of the experiment in (Campbell, 1997). Since the best results were gained from using LSP frequencies and the Divergence Shape metric I will discuss these only, although substituting one of the other sets of features or the Bhattacharya distance measure would be trivial.

5.1.3 Enrolment and Identification

As mentioned in chapter 2, there are two main stages in a speaker identification system, which are enrolment and identification.

In the enrolment stage, the training utterances are each analyzed using signal processing techniques to yield several LSP feature vectors for each utterance. All the feature vectors that are computed for each utterance are combined to create a speaker model. This model then represents the target person. The system in this experiment is an open system, which means that it is possible for people to make use of the system who haven't enroled. Therefore an unknown test speaker is not guaranteed to have an existing voice model. In (Campbell, 1997), target speakers and test speakers are randomly selected from a larger set of speakers and are compared, which may result in a comparison of a test utterance coming from a speaker that does not have a speaker model present in the set of target speaker models.

In the identification stage, an input utterance is obtained from an unknown person. Feature vectors containing the LSP frequencies are extracted from the input utterance and are used for comparison with each of the speaker models that are stored in the system in order to determine which model the utterance is closest to. The utterance is then classified as having come from the speaker to which the model belongs.

In this experiment, 44 people were used for training and testing. Each of the four training sessions were used for training, and each of the ten testing sessions were used for

identification. A one-to-many matching procedure is used to compare a test utterance against every voice model in the database, and the closest match is considered the identity of the unknown test speaker.

5.1.4 Speech Processing Procedure

The speech processing procedure will be explained from start to finish in order to illustrate an example of how it is done in general.

5.1.4.1 Signal Processing

To process an input speech waveform, the signal is first broken up into adjacent, overlapping frames. Each frame has a duration of 20ms and adjacent frames overlap by 10ms.

A voicing analysis was performed in (Campbell, 1997) on each frame to determine whether the signal segment within that frame is voiced or not. Note that this voicing analysis was performed specifically in the system in (Campbell, 1997) but is not necessarily a standard procedure in general. In their experiment, only frames that contain voiced data were used for analysis and modeling because voiced portions of speech contain more highly speaker-discriminatory information. As mentioned in chapter 2 in the section on syllables, the most sonorant portion of the syllable is the nucleus, and thus this would be the portion that would contain the most speaker-specific information.

5.1.4.2 Frame Windowing

Frames to be used in the speech analysis are windowed using a Hamming window. As mentioned in chapter 3, windowing is necessary to taper off the signal due to discontinuities that occur at the endpoints of the signal segment. These discontinuities are undesirable for signal processing analysis in the frequency domain. Once a frame has been windowed, it is ready for any signal processing and feature extraction that needs to be performed on it.

5.1.5 Features

An autoregressive computation is performed on the windowed signal within these voiced frames to obtain Linear Prediction Coefficients (LPC). In Campbell's paper, the bandwidth of the formants is expanded by 15Hz. This is achieved by multiplying each term by a value $0 < \gamma < 1$. In this case (Campbell, 1997):

$$\gamma = 0.994 \tag{5.1}$$

This effectively shifts the poles of the all-pole transfer function of the autoregressive LPC model radially towards the origin. Next, the LPC coefficients are converted to Line Spectrum Pair (LSP) frequencies, which are used to create the voice models that are used for the modeling and classification of each speaker.

The LPC features were computed using the Durbin-Levinson recursion (Campbell, 1997), which computes not only the final linear prediction coefficients but also the intermediate

reflection coefficients. These intermediate coefficients can be used as a feature themselves and are considered to be another alternative representation of the LPC features, but are not considered in this thesis.

The LPC coefficients were converted to Line Spectrum Pair Frequencies. The algorithm for converting LSP Frequencies is given in (Soong and Juang, 1984) and is mentioned in chapter 4.

5.1.6 Speaker Modeling and Classification

As already mentioned, the Divergence Shape classifier, which is described in chapter 4, was used for modeling and classification. Essentially, a covariance matrix and mean vector is computed for the training utterances for a given speaker, which represents the speaker model. Classification is performed by computing a distortion (Divergence Shape) between two covariance matrices and their respective mean vectors.

5.2 Previous Experiments

Prior to the main experiment that is the focus of this chapter that tests the novel PLP-LSP technique, I performed several other experiments in order to get a feel for some of the different features and methods that have been mentioned throughout this thesis and within the experiment in (Campbell, 1997) in particular.

Some of my earlier experimentation includes text-independent speaker identification, text-independent speaker verification, and text-independent speech identification

(phoneme identification). The first two experiments were done using LSP frequencies and using the TIMIT speech corpus, which contains spontaneous conversational speech over the telephone, and was created primarily for speech recognition purposes, as phonetic and word segmentation have been performed on it. However, it can also be used for speaker recognition.

I started by trying to reproduce the procedure in (Campbell, 1997). I tested the three features (Log Area Ratios, Reflection Coefficients, and Line Spectrum Pair frequencies) using both the Divergence and Bhattacharyya methods. I tried the Divergence measure and compared it to the Divergence Shape component, and did the same for the Bhattacharyya measure. I found that the Divergence measure outperformed the other methods, including the Divergence Shape measure. I eventually switched to Vector Quantization. I had used this classifier for a class project and found it to perform better than the Divergence measure. I, too, found that LSP frequencies were the best choice of feature due to higher recognition rates. I then switched from identification to verification using the Vector Quantization with LSP. Both speaker recognition experiments had similar results to each other and to (Campbell, 1997). As already mentioned, (Campbell, 1997) had an identification accuracy of around 98.9%.

The speech recognition experiment was done comparing PLP with LPC, and then also testing PLP-LSP, using the TIMIT speech corpus. While the phoneme identification accuracies were quite low (probably due to the simplicity of the experiment), the PLP-LSP features outperformed the PLP features, which was what I was trying to discover in

the course of that particular experiment. These results proved promising, which prompted me to test the novel PLP-LSP features in a speaker recognition task.

5.3 PLP-LSP Experiment

In this experiment, which is the focus of this chapter (and this thesis), I use Joe Campbell's Divergence Shape recognition experiment as a template for comparison of the four features mentioned. However a VQ classifier similar to the one described in (Pop and Lupu, 2002) was used in lieu of the Divergence classification method since the VQ method yielded better results in the evolved experiments that I performed in the past. My VQ classifier yielded very similar results to Joe Campbell's Divergence shape classifier for the LSP features. My experiment yielded 98.2%, whereas Joe Campbell's experiment yielded 98.9%, using the YOHO speech corpus.

In addition, the order of the features was increased to match that commonly used in the literature for PLP coefficients. This experiment was performed in order to explore how well each of the four features performs on a text-independent identification task. Text-independent speaker recognition is much more difficult a task than text-dependent speaker recognition because the classifier must be trained to recognize the person from any arbitrary utterance from the target speaker.

In this chapter, a description of the experiment will be given. Because an overview of the steps taken to perform the speech analysis as described in chapter 3 was given as an example of a typical speech processing pipeline in my description of the experiment

performed by (Campbell, 1997), these steps will be omitted here. Even though LSP frequencies were used in that description, any features can (and will) be substituted in their place. Because the LPC, LSP, and PLP features have already been discussed in chapter 3, the details of their computation are omitted here. The computation of the PLP-LSP features is discussed briefly.

5.3.1 Speech Database

The YOHO speech corpus (Campbell and Reynolds, 1999), (http://wave.ldc.upenn.edu/Catalog/readme_files/yoho.readme.html) was used for this experiment. As mentioned, (Campbell, 1997) used only 44 of the 138 speakers for the speaker recognition task, as will be the case in this experiment.

5.3.2 Overview

The purpose of this section is not only to describe the text-independent speaker identification method used in this experiment but also to illustrate each of the steps in the classification pipeline, such as framing, frame windowing, feature extraction, and modeling and classification.

5.3.3 Enrolment and Identification

In this experiment, 44 people were used for training and testing. Each of the four training sessions were used for training, and each of the ten testing sessions were used for identification. A one-to-many matching procedure is used to compare a test utterance

against every voice model in the database, and the closest match is considered the identity of the unknown test speaker. The speaker identification task was performed four separate times, each time using one of the four features. The resulting identification accuracies are tabulated and discussed in chapter 6.

5.3.4 Speech Signal Processing

As in (Campbell, 1997), the signal is first broken up into adjacent, overlapping frames. Each frame has a duration of 20ms and adjacent frames overlap by 10ms.

The voicing analysis they used is omitted from my experiment. I chose not to use the voicing decision in the end because it only improves the recognition by a marginal amount, as suggested by J. Campbell (personal communication, August 4, 2002). So, all frames are used in the analysis.

5.3.5 Features

The four features used in this experiment are LPC, LSP, PLP, and the novel PLP-LSP features. Each of the features is of order 20. According to (Hermansky, 1990), lower-order PLP features preserve the lexical content of the speech while ignoring, to a large extent, the speaker-dependent voice characteristics found within the speech. Higher orders contain much more speaker-specific information, which is what is required for speaker recognition applications. It is common to use an order as high as 20 for PLP.

In the case of LSP frequencies, an autoregressive computation is performed on the windowed signal within these voiced frames to obtain Linear Prediction Coefficients (LPC). Recall that in Campbell's paper, the bandwidth of the formants is expanded by 15Hz. This is achieved by multiplying each term by a value $0 < \gamma < 1$, as in equation 5.1. This step was not performed in my experiment because I didn't want to introduce an added layer of complication to the LPC coefficients. I wanted to keep them as raw as possible so as not to introduce more variables into the experiment that might confound the results.

The LPC coefficients were computed using the Levison-Durbin algorithm, and then converted to LSP frequencies using the technique in chapter 4.

In order to derive the PLP-LSP features, the PLP features are transformed to LSP frequencies using the same procedure as that used on the LPC features. A description of the computation of the PLP coefficients is given in chapter 4. Because the last step of the PLP computation is an autoregressive calculation, much like the computation of the LPC features, this transformation to the LSP representation in the context of PLP features makes sense.

5.3.6 Speaker Modeling and Classification

As already mentioned, the Divergence classifier was replaced with a VQ codebook because the VQ method yielded higher accuracies in this experiment. The following is a description of the modeling and classification procedure used in this experiment.

5.3.6.1 Overview

Using information from (Pop and Lupu, 2002) and (http://www.data-compression.com/vq.html), I constructed a single-section vector quantization speaker verification system using the LBG training algorithm (http://www.data-compression.com/vq.html) and used a one-to-many matching scheme on 44 speakers from the YOHO database. Although there are several methods that can be used to construct a codebook for a speaker, such as self-organizing maps (http://www.cis.hut.fi/projects/ide/publications/html/mastersJV97/node3.html) or simulated annealing (Lu and Morrell, 1991), it has been found that there is only a marginal difference in accuracy between the codebooks constructed using each of the methods. I used the LBG algorithm, which is described in (http://www.data-compression.com/vq.html) since it is the simplest. Each LSP feature vector that is extracted from an input speech waveform is used to train the codebook for a speaker. The codebook with the smallest average distortion to each input feature vector (using a Euclidean distance measure) is the codebook that is used as the speaker model for that person. To test an unknown person against a target speaker, an input test utterance is obtained and the same feature extraction technique is used to obtain a set of LSP frequency feature vectors for the unknown person. These feature vectors are encoded in a speaker's codebook and the average distortion measure is computed. This value is then compared against the distortion measure of every other speaker's codebook, and the identity of the speaker with the codebook having the smallest distortion is deemed to be the identity of the unknown test speaker.

CHAPTER SIX: RESULTS AND CONCLUSIONS

The experiment in chapter 5 was performed to determine two things. First it was performed to see if the new PLP-LSP features outperform the traditional PLP features in a text-independent speaker identification task. Second it was performed to see how the new PLP-LSP features compare to the traditional LSP features computed directly from LPC features. Specifically, since PLP features outperform LPC features, do PLP-LSP features also outperform LSP features?

Table 6.1 shows the percent correct identifications for 44 people from the YOHO database using 20th-order LPC, LSP, PLP, and PLP-LSP features.

Table 6.1: Percent correct identification for the four features from the experiment in chapter 5.

LPC	PLP	LSP	PLP-LSP
88.737	93.977	98.182	96.08

6.1 PLP-LSP versus PLP

Clearly, the PLP-LSP features outperform the LPC and PLP features in terms of identification accuracy. Therefore the alternative LSP representation of the PLP method was a success. The benefits of the LSP frequencies have been extended to the PLP features. The new PLP-LSP method is a more robust psychoacoustic choice for text-

independent speaker recognition. These results are very encouraging, and ultimately it is likely that the PLP-LSP method will be a better choice than PLP for other speaker and speech recognition tasks as well. It is also likely that the LSP transformation will work for various other features with autoregressive components, hopefully yielding increases in recognition performance in various different areas of speech analysis. It is unclear whether LSP frequencies can be computed for other features in general, such as those that do not share the autoregressive property seen in LPC and PLP features. Can they be massaged into a form in which such a transformation makes sense? If so, perhaps they too can benefit from the enhanced properties of the LSP frequency representation.

6.2 PLP-LSP versus LSP

When compared to the traditional LSP features, the PLP-LSP features performed worse. However, it is uncertain whether the traditional features are more able to adequately model the speaker-discriminative properties of speech outside of this experiment. The two are very close and more rigorous testing would have to be performed to determine if one is indeed better than the other.

6.3 Conclusions and Future Work

6.3.1 Statistical Significance

To compute the statistical significance of the difference between the accuracies of the PLP-LSP vs PLP and PLP-LSP vs LSP features, I used Student's two-tailed t-Test. The test is designed to compare two population means based on their distributions at some

level of confidence. Normally a confidence level of 95% or 99% is used. A test statistic is computed between the two populations and this value is compared to the value in a lookup table. The appropriate table row is found by computing the Degrees of Freedom (df) and is found in the column corresponding to the chosen level of confidence.

The test statistic is computed as follows:

$$t = \left| \frac{(mean_{PLP-LSP} - mean_{PLP})\sqrt{n}}{stdDev_{PLP-LSP}^{\ 2} + stdDev_{PLP}^{\ 2}} \right| \tag{6.1}$$

Where n is the number of speakers. The df is computed as:

$$df = \left\lfloor \frac{[(s_1^{\ 2}/n_1) + (s_2^{\ 2}/n_2)]^2}{\frac{(s_1^{\ 2}/n_1)^2}{n_1 - 1} + \frac{(s_2^{\ 2}/n_2)^2}{n_2 - 1}} \right\rfloor \tag{6.2}$$

In this case, the number of speakers for each population, n_1 and n_2, are the same. Given that we chose a confidence level of 95%, if the test statistic is less than the value in the lookup table, the difference between the means is said to be insignificant, with 95% confidence. This means that there is also a 5% chance that the difference is actually significant. On the other hand, if the test statistic is greater than the value in the lookup table, the difference is said to be significant, statistically speaking.

In my experiment I have found the difference in accuracies of the PLP and the PLP-LSP features to be insignificant at 95% confidence using Student's T-test. On the other hand, the difference in accuracies of the PLP-LSP and the LSP features was found to be significant at 95% confidence. The numbers used to determine the significance of the difference between the accuracies of each is as follows:

For the case of PLP-LSP versus PLP, given that:

$mean_{PLP-LSP} = 96.0795$

$stdDev_{PLP-LSP} = 7.01737$

and

$mean_{PLP} = 93.9773$

$stdDev_{PLP} = 8.86458$

In our case n is equal to 44, and t was found to be:

$t = 1.23341$

with

$df = 81$

This is lower than the t-value of the table for 81 degrees of freedom at 95% confidence which is t=1.664 (Weiss, 2004) (pg. A-11), which means that the difference between PLP-LSP and PLP is not significant.

For the case of PLP-LSP versus LSP, given that:

$mean_{PLP-LSP} = 96.0795$

$stdDev_{PLP-LSP} = 7.01737$

and

$mean_{LSP} = 98.1818$

$stdDev_{LSP} = 3.97535$

then t is computed as:

$t = 1.72903$

with

$df = 68$

This is higher than the t-value of the table for 68 degrees of freedom at 95% confidence which is t=1.668 (Weiss, 2004) (pg. A-11), which means that the difference between PLP-LSP and LSP is significant.

I found this result surprising. I expected, or hoped, that the opposite would be true. That is, I was hoping that the difference between the PLP-LSP features and the PLP features would be significant, and the difference between the PLP-LSP features and the LSP features would be insignificant.

6.3.2 Conclusions

6.3.2.1 Benefits of PLP-LSP

One might ask what the benefit of the PLP-LSP features is if it isn't significantly better than the PLP features and also can't outperform the already well-known LSP features. The first, and most important point, is that this experiment was performed primarily to see if the LSP representation could even be computed for the PLP features, and if so, to see if any improvement was gained over the original representation from using the novel LSP representation. This experiment has shown that both of these things are possible. So it is at least a step in the right direction. Discovering what relationship PLP-LSP had with respect to LPC and LSP was only a secondary goal, to see where it fit into the known picture.

Secondly, PLP is only one of several psychoacoustic features. There are other psychoacoustic features that seem to do better than PLP, and also have the autoregressive property. Essentially, these psychoacoustic features all share a common set of steps, but with minor differences, such as the shape of the auditory filters in the filterbank. So, given that the "PLP-LSP is better than PLP" relationship is established, it is highly likely that there exist other psychoacoustic features that probably do outperform LSP, given that they have been converted to their LSP frequency representation themselves. This is yet to be seen, but is obviously a plausible avenue to explore in the future in light of my results.

Third, there is no reason to believe that there aren't other, non-psychoacoustic features that can be enhanced in this fashion. The only reason a psychoacoustic feature was selected was because of the fact that it tries to simulate the behavior of the human ear, which already performs speaker recognition very well. So the same argument holds for these features.

And finally, PLP is generally a better option than LPC for *speech* recognition (Hermansky, 1990) as well as *speaker* recognition (http://www.asel.udel.edu/icslp/cdrom/vol3/706/a706.pdf), only more so (Hermansky, 1990). In speech recognition, much lower feature vector orders are used (say, around 8 or less) because the speaker-specific information tends to be suppressed at these levels. So, while PLP-LSP wasn't significantly different in terms of accuracies from LSP in speaker recognition, this may not be the case in speech recognition. Perhaps the gap between PLP and LPC would widen even more, and the PLP-LSP might surpass the LSP.

Clearly, the experiment performed to explore the primary objective, which was to find out whether PLP-LSP outperforms PLP, and the secondary objective, which was to find out whether PLP-LSP outperforms LSP, has raised more questions, and the results are by no means conclusive. Had the PLP-LSP actually outperformed LSP, some of these questions would still need to be addressed. Much more research needs to be done in order to find the answers and what magnitude they have. This, however, is not in the scope of this thesis, and so I leave it to future research.

In addition to the above points, I would like to discuss the possibility of using the PLP-LSP and LSP features together in a multiple classifier approach. It is a combination that would likely yield significantly higher success rates.

6.3.2.2 Multiple Classifier Approach

In the case where we have several non-homogeneous classifiers, each of which gives differing classification results, it is possible to combine these methods in a multiple classification scheme. The idea is that because the properties of each classifier are different, where one classifier fails, the other classifiers may succeed. Thus a scheme can be employed to determine the combined verdict of several classifiers, such as a majority vote, borda count, sum rule, or other (Parker, 2001), even using only two classifiers. The idea of a combined classifier scheme is that its overall performance should be higher than that of any individual classifier that it is made up of, hence generating a more powerful classification scheme. To determine whether the LSP and PLP-LSP classifiers are non-homogeneous and to what extent, I performed the following computation: for each speaker, I compared the classification result of the LSP classifier with that of the PLP-LSP classifier for every test utterance. A count was taken of the number of times the LSP classifier classified correctly but the PLP-LSP classifier classified incorrectly, and vice versa. The results are shown in table 6.2 below.

Table 6.2: Percent of Time One Method Outperforms the Other

PLP-LSP	LSP
17.4603	82.5397

This table shows how often PLP-LSP correctly classified an utterance for a speaker when LSP did not, and vice versa. The numbers are a percent of the total number of times one classifier was correct and the other was not. The table shows that there are clear differences in the classification outcomes of each of the classifiers, which means that these are likely good candidates for use in a combined classifier scheme and would probably significantly increase the success rate.

Bibliography

Bekesy, G. von (1942). Uber die Schwingungen der Schneckentrennwand beim Präparat und Ohrenmodell, *Akust. Zeits.*, *7*, 173-186.

Boite, J. M. & Ris, C. (1999). Development of a French Speech

Burton, D. (1985, April). Applying matrix quantization to isolated word recognition. *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP-85), 10,* Naval Research Laboratory, Washington, D.C., 29-32.

Campbell, J. & Reynolds, D. (1999, May 15-19). Corpora for the Evaluation of Speaker Recognition Systems. *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP-99)*, Phoenix, Arizona, 2247-2250.

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

Campbell, J. P.& Tremain, T. E. (1986). Voiced/Unvoiced Classification of Speech with Applications to the U.S. Government LPC-10E Algorithm. *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 473-476.

Campbell, J. P., Tremain, T. E. & Welch, V. C. (1990, April/May). The Proposed Federal Standard 1016 4800 bps Voice Coder: CELP. *Speech Technology Magazine*, pp. 58-64.

Campbell, J. P., Tremain, T. E. & Welch, V. C. (1991). The Federal Standard 1016 4800 bps CELP Voice Coder. *Digital Signal Processing, Academic Press, 1*(3), 145-155.

Egan, J. P., Hake, H. W. (1950). On the Masking Pattern of a Simple Auditory Stimulus. *Journal of the Acoustical Society of America, 22*, 622-630.

Gaswami, J. C. & Chan, A. K. (1999) *Fundamentals of Wavelets: Theory, Algorithms, and Applications*. New York: John Wiley & Sons, Inc.

Goldstein, E. B. (1999) *Sensation and Perception* (5th ed.). Pacific Grove, California: Brooks/Cole Publishing Company.

Greenberg, S. & Kingsbury, B. E. D. (1997). The Modulation Spectrogram: In Pursuit of an Invariant Representation of Speech. *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP-97)*, 1647-1650.

Greenberg, S. (1998) Speaking in Shorthand – a Syllable-Centric Perspective for Understanding Pronunciation Variation. *Proceedings of the ESCA Workshop on Modeling Pronunciation Variation for Automatic Speech Recognition*, Kekrade, Netherlands, 47-56.

Hermansky, H. (1990, April). Perceptual Linear Predictive (PLP) Analysis of Speech. *Journal of the Acoustical Society of America, 87*(4), 1738-1752.

Higgins, A., Bahler, L. & Porter, J. (1991). Speaker Verification Using Randomized Phrase Prompting. *Digital Signal Processing, (1)*, 89-106.

Huang, C., Chen, T. & Chang, E. (2002). Adaptive Model Combination For Dynamic Speaker Selection Training. *Proceedings of the International Conference on Spoken Language Processing (ICSLP-2002), 1*, Denver, USA. 65-68.

Itakura, F. (1975, April). Line Spectrum Representation of Linear Predictive Coefficients of Speech Signals. *Journal of the Acoustical Society of America, 57*, p. S37.

Kailath, T. (1967, January). The Divergence and the Bhattacharyya Distance Measures in Signal Selection. *IEEE Trans. Commun. Technol., COM-15*, 52-60.

Kasuriya, S., Wutiwiwatchai, C., Achariyakulporn, V., & Tanprasert, C. (2001). Comparative Study of Continuous Hidden Markov Models (CHMM) and Artificial Neural Network (ANN) on Speaker Identification System. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems, 9*(6), 673-683.

Keogh, E. J. & Pazzani, M. J. (2000, August 20-23). Scaling up Dynamic Time Warping to Massive Datasets. *Proceedings of the 6th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Boston, Massachusetts, 285-289.

Kingsbury, B. E. D., Morgan, N. & Greenberg, S. (1998) Robust Speech Recognition using the Modulation Spectrogram. *Speech Communication, 25*, 117-132.

Lawrence, S., Burns, I., Back, A. D., Tsoi A. C. & Giles C. L. (1998). *Neural Network Classification and Prior Class Probabilities*. Tricks of the Trade, Lecture Notes in Computer Science State-of-the-Art Surveys, Orr, G. Mueller & K-R., Caruana, R. (Eds.), Springer-Verlag, 299-314.

Lu, N. A., Morrell, D. R. (1991, April 14-17). VQ Codebook Design using Improved Simulated Annealing Algorithms. *International Conference on Acoustics, Speech, and Signal Processing (ICASSP-91), 1*, Arizona State University, Tempe, Arizona, 673-676.

Mahadeva Prasanna, S. R., Gangashetty, S. V. & Yegnanarayana, B. (2001, July 15-18). Significance of Vowel Onset Point for Speech Analysis. *Sixth Biennial Conference on Signal Processing and Communications*, Bangalore, India, 81-88.

Makhoul, J. (1975, April). Linear Prediction: A Tutorial Review. *Proceedings of the IEEE, 63*(4), 561-580.

Makhoul, J. (1975, June). Spectral Linear Prediction: Properties and Applications. *IEEE Trans. Acoust, Speech, Signal Processing, ASSP-32*(3), 283-296.

Martens, J. P., Binnenpoorte, D., Demuynck, K., van Parys, R., Laureys, T., Goedertier, W. & Duchateau, J. (2002, January). Word Segmentation in the Spoken Dutch Corpus. *Proceedings of Language Resources and Evaluation (LREC-2002), 14*(1), Las Palmas de Gran Canaria, Spain, 23-35.

Meinedo, H. & Neto, J. P. (2000). The Use of Syllable Segmentation Information in Continuous Speech Recognition Hybrid Systems Applied to the Portuguese Language. *Proceedings of the International Conference on Spoken Language Processing (ICSLP-2000)*, Beijing, China.

Meinedo, H., Neto, J. P. & Almeida, L. B. (1999). Syllable Onset Detection Applied to the Portuguese Language. *Proceedings EUROSPEECH-99*, Budapest, Hungary, 81-84.

Moon, T. K. (1996, November). The Expectation-Maximization Algorithm. *IEEE Signal Processing Magazine*, *13*(6), 47-60.

Morgan, N. & Gold, B. (2000) *Speech and Audio Signal Processing :Processing and Perception of Speech and Music.* New York: John Wiley & Sons, Inc.

Nakamura, S. & Markov, K. (2004, January 12-14). A Hybrid HMM/Bayesian Network Approach to Robust Speech Recognition. *Proceedings of the Special Workshop in MAUI (SWIM)*, Maui, Hawaii, CD-ROM.

Nemer, E., Goubran, R. & Mahmoud, S. (1997, September). Voicing Decision and Pitch Estimation using Third-Order Cumulants. *ICSPAT-97*, San Diego, CA.

Oppenheim, A. V. & Schafer, R. W. (1989) *Discrete-Time Signal Processing*, Upper Saddle River, New Jersey: Prentice Hall.

Parker, J. R. (2001). Rank And Response Combination From Confusion Matrix Data, *Information Fusion*, 2, 113-120.

Patterson, R. D. (1976). Auditory Filter Shapes Derived with Noise Stimuli. *Journal of the Acoustical Society of America, 59*, 640-654.

Pop, P. G. & Lupu, E. (2002) *Speaker Verification with Vector Quantisation* Retrieved (Jan, 01, 2005) from http://193.226.6.174/IT2002/pdf/L28.pdf

Rabiner, L. R. & Juang, B. H. (1986). An Introduction to Hidden Markov Models. *IEEE ASSP Magazine, 3*(1), 4-16.

Rabiner, L. R. (1989). A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. *Proceedings of the IEEE, 77*(2), 257-286.

Ranke, O. F. (1950). Hydrodynamik der Schneckenflüssigkeit. *Z. Biol., 103*, 409-416.

Recognizer using a Hybrid HMM/MLP System. *Proceedings of European Symposium on Artificial Neural Networks (ESANN)*, 441–446.

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

Ribeiro, C. M. & Trancoso, I. M. (1996, September). Application of Speaker Modification Techniques to Phonetic Vocoding. *International Conference on Speech and Language Processing (ICSLP'96), 1,* Philadelphia, 306-309.

Rodríguez, E., Ruíz, B., García-Crespo, Á. & García, F. (1997). Speech/Speaker Recognition Using a HMM/GMM Hybrid Model. *AVBPA-1997*, 227-234.

Schroeder, M. R. (1977). Recognition of Complex Acoustic Signals. *Life Sciences Research Report, 55*, 323-328.

Shastri, L., Chang, S. & Greenberg, S. (1999, August) Syllable Detection and Segmentation Using Temporal Flow Neural Networks. *Proceedings of the Fourteenth International Congress of Phonetic Sciences, 3*, San Francisco. 1721-1724.

Soong, F. K. & Juang, B. H. (1984). Line Spectrum Pair (LSP) and Speech Data Compression. *Proc ICASSP*, 1.10.1-1.10.4.

Stark, H. & Woods, J. W. (2002). *Probability and Random Processes with Applications to Signal Processing* (3rd ed.). New Jersey: Prentice-Hall.

Stolcke, A., & Shriberg, E. (1996, October). Automatic Linguistic Segmentation of Conversational Speech. *Proceedings of the International Conference on Spoken Language Processing (ICSLP-96)*, 1005-1008.

Vandecatseye, A. & and Martens, J. P. (2003, September). A Fast, Accurate and Stream-Based Speaker Segmentation and Clustering Algorithm. *Proceedings of the 8th European Conference on Speech Communication and Technology, 2*(14), Geneva, Switzerland, 941-944.

Weiss, N. A. (2004) *Introductory Statistics* (7th ed.). Boston, Massachusetts: Addison-Wesley.

Xiang, B., Chaudhari, U. V., Navratil, J., Ramaswamy, G. N. & Gopinath, R. A. (2002, May). Short-time Gaussianization for Robust Speaker Verification. *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP-2002), 1*, Orlando, Florida, 681-684.

Yun, Y. & Oh, Y. (2000, June). A Segmental-Feature HMM for Speech Pattern Modeling. *IEEE Signal Processing Letters , 7*(6), 135-137.

Zheng, F., Song, Z., Li, L., Yu, W. & Wu, W. (1998). The distance measure for Line Spectrum Pairs applied to speech Recognition. *Proceedings of the International Conference on Spoken Language Processing (ICSLP-98), 3,* 1123-1126.

Zwicker, E. (1970). Masking and Psychological Excitation as Consequences of the Ear's Frequency Analysis. R. Plomp & G. F. Smoorenburg (Eds.), *Frequency Analysis and Periodicity Detection in Hearing*, Sijthoff, Leiden, The Netherlands, 376-396.

Zwisloki, J. J. (1948). Theorie der Schneckenmechanik. *Acta Oto-Laryngol., Suppl., 72,* 3-26.