Abstract

WANG, MENG. Development of Digital Signal Processing and Statistical Classification Methods for Distinguishing Nasal Consonants. (Under the direction of David McAllister.)

For almost half a century, people have been looking for efficient classifiers to distinguish two nasal sounds, /*m*/ from /*n*/, uttered by a single speaker. From the middle of the last decade, there has been little progress in research on this topic. In recent years, we, researchers of the Voice I/O Group in Department of Computer Science at North Carolina State University, have conducted some new trials on this classical problem. In this thesis, those trials are briefly summarized. Instead of simply using the Fourier transform to produce the spectra as people usually did in the past, the author uses other kinds of transforms to extract more feature differences between /*m*/ and /*n*/. The new transforms can be the alternatives of frequencies, such as singular values or eigenvalues, or even other transforms such as wavelets, which can deal with non-stationary systems quite well. We combine together the old and new features to get a larger feature vector, which will bring more classification information. We collect multiple voice samples of a single speaker and calculate the above feature representations, then use them as input of some popular statistical classification techniques, such as Principle Component Analysis (PCA), Discriminant Analysis (DA), and Support Vector Machine (SVM). By way of one training process, one testing process, and one heuristic scheme, we can identify the nasals with low error rates.

Development of
Digital Signal Processing and Statistical Classification Methods for Distinguishing Nasal Consonants

By
Meng Wang

A thesis submitted to the Graduate Faculty of
North Carolina State University
in partial fulfillment of the
requirement for the Degree of
Master of Science

Operations Research

Raleigh

2003

APPROVED BY:

_____ _____

Chair of Advisory Committee

PERSONAL BIOGRAPHY

Meng Wang was born on February 26, 1975 in Suxian City, P.R.China. He started school when he was six. In the same city, he continued his education until he graduated from high school in 1992.

Meng then went to Nanjing City in Jiangsu Province to attend college in Southeast University. And in May 1996, he graduated with honors, receiving a Bachelor of Science degree in Mathematics, with a concentration in Applied Mathematics. He was also approved to be a candidate of a Master of Science in the same department without taking any entry tests.

He continued academic study in Southeast University and graduated with the Master degree in May 1999. Then he came to the United States to enter a Ph.D program in Operations Research Program. From February 2001, he began to work with the Voice I/O Group on the Lip Sync Project. From September 2001, he started working on nasal identification problem.

In March 2003, Meng stopped his Ph.D study and transfered to a thesis Master student. He is scheduled to graduate in August 2003.

Meng's main interests include web searching, reading and anything related to computer software. He also has interest on driving and sports, such as chess, soccer and badminton.

TABLE OF CONTENTS

LIST OF TABLES

Page

LIST OF FIGURES

LIST OF NOTATIONS AND SYMBOLS

Notations

Tables, figures and equations in each chapter are numbered serially. The first number indicates the chapter number, and the second number indicates the table, figure or equation number. The two numbers are separated by a dot. If a reference is made to, say equation number (3.2) in a chapter, it means equation 2 of chapter 3. Vectors are in general column vectors under lower-case; matrices are under capital-case.

Symbol Definition

Symbols are listed using alphabetical order for lower case, upper case, Arabic and Greek formats. The page numbers where the symbols are used for the first time are also indicated.

[1] a : scale-factor 23

[2] cm_2 : second central moment 16

[3] d : dimension of feature vectors 26

[4] $\tilde{d}$, d' : dimension of feature vectors 69

[5] $\exp(\cdot)$: exponential function 64

[6] $fm = (fm_1, fm_2, ..., fm_{44})$: mean vector 27

[7] $freq$: frequency in Hertz 16

[8] $fstd = (fstd_1, fstd_2, ..., fstd_{44})$: standard deviation 27

[9] l_m : length of the side-branching oral cavity with a unit of centimeter 12

[10] $/m/$, $/n/$: nasal sounds 1

[11] m_1 : first moment 16

[12] $p(\cdot)$: probability density function 16

[13] q_i^L , q_i^R : feature vectors 57

[14] $q^{PCA,L}{}_i$, $q^{PCA,R}{}_i$: PCA-derived vector 57

[15] r : Mahalanobis distance 29

[16] r_i : Mahalanobis distance 77

[17] x : feature vector 26

[18] $\hat{x}$, $\tilde{x}$: feature vectors 72

[19] y , y_i : LDA - transformed training data 77

[20] A_i : transformation matrix 77

[21] B : Traunmüller's approximation for bark scale 24

[22] C , C_i , C_1 , C_2 : covariance matrix 28

[23] $\hat{C}$: covariance matrix 70

[24] C^T : transpose of C 26

[25] $|C|$: determinant of C 64

[26] C_{sound} : speed of sound 11

[27] $E(\cdot)$: expectation values 26

[28] FFT: Fast Fourier Transform 27

[29] F_s : Sampling rate 15

[30] $GACV$: the Generalized Approximate Cross Validation 79

[31] $GCKL$: the Generalized Comparative Kullback-Liebler distance 80

[32] GP : glottal pulse 15

[33] H_k : The Reproducing kernel Hilbert space (RKHS) 80

[34] $H(z)$: Transfer function 7

[35] $I(n)$: input signal 23

[36] $K(p,t)$: Reproducing kernel 80

[37] L: total length of the vocal tract 11

[38] M: mass: 16

[39] N1, N2, N3: nasal formants 11

[40] N_1, N_2: numbers 28

[41] $O(n)$: output signal 23

[42] $P(\cdot)$: probability 62

[43] $Pitch$: Pitch 15

[44] $S(freq)$: spectrum 16

[45] S_w, S_b: covariance matrix 75

[46] V: transformation matrix 75

[47] V^L, V^R: transformation matrix 57

[48] z-transform of the input signal: $X(z)$ 7

[49] z-transform of the output signal: $Y(z)$ 7

[50] $\square_1, \square_2, \ldots, \square_k$: eigenvectors 26

[51] $\square_i$, $\square'_i$: training data feature vector 27

[52] $\square^{PCA}{}_i$: PCA-derived vector 28

[53] $\square$, $\square'$: testing data feature vector 27

[54] $\square^{PCA}$: PCA-derived vector 28

[55] $\square$: PCA-derived vector 26

[56] $\square_1$, $\square_2$: means 55

[57] $\square_3$: mean 75

[58] $\square_i(x)$: Discriminant function 64

[59] $\square_{ij}$: correlation coefficient 68

Chapter 1

Introduction and Background

Distinguishing between two nasal consonants, /*m*/ and /*n*/, has been a classical problem of acoustic phonetics for half a century. The progress of this topic was well summarized in [1]. Nearly all the work in this area suggests that the information needed to solve this problem lies in the shape of the various spectra of the nasal consonants themselves as well as the surrounding sounds, including the transitions to and from the nasals. Phoneticians for years have extracted some useful information, such as the different locations of the anti-formants and the formants between /*m*/ and /*n*/, from the spectrum. These differences are reflected in the shape of the various spectra. In addition, the phoneticians have observed that the starting frequency of the second formant provides a cue for distinguishing /*m*/ from /*n*/, again a matter of spectral shape. In 1994, Harrington proposed the Combined Spectra Method [1], which uses the spectral information from both the nasal sounds and the vowel sounds around the nasals. The Combined Spectra Method shows the best performance on identifying /*m*/ and /*n*/ until recent years, although the accuracy is no more than 94% for both syllable-initial (nasals precede the vowels) and syllable-final (nasals succeed the vowels) cases. To better understand this important method, we duplicate and discuss it briefly in Chapter 3. However, in Harrington's experiments, the data were collected for multiple speakers. In this thesis, we are conducting experiments for one single speaker.

In the past few years, researchers in the Voice I/O group in the Department of Computer Science at North Carolina State University have made some efforts to extract differences between /*m*/ and /*n*/ using a 2-D moment space of the spectra. This method was initially used to determine *visemes* (visible phonemes) for different classes of sounds for a project named "A Fully Automated System of Spontaneous Speech Lip Synchronization**.**"[96]. Moment-space classifiers have proven themselves for other classes of sounds such as vowels, but it did not work well to differentiate nasals /*m*/ and /*n*/. Those two nasal sounds are

acoustically very similar while corresponding to vastly different visemes. More specifically, the lips are closed for / *m* /, and open for / *n* /.

The remainder of this thesis is organized as follows. Chapter 1 provides the background of this study, which contains a brief description of the phonetic and acoustical properties of nasals. Chapter 2 presents an overview of past efforts on this topic. Chapter 3 duplicates the Combined Spectra Method and re-evaluates its performance. Chapter 4 sheds light on some new approaches, such as eigenvalue tests and singular-value tests, that are used to produce more feature vector components. Chapter 5 uses some efficient statistical classification methods to process the features derived from the Combined Spectra Method, singular-value or eigenvalue test, moment-space method, root-mean-square values, and the locations of anti-formants and formants. Chapter 6 gives the conclusions and topics of future research.

1.1 Speech Production

Overview

Normal speech sounds are produced by modulating an outward flow of air. For most sounds, the lungs furnish the stream of air, which flows between the vocal folds (cords), and causes them to vibrate, thereby modulating the air. The air then passes through several vocal cavities before it exits from the body through the mouth and to a slight degree through the nostrils. Speech sounds produced in this way are called *voiced sounds*.

Sounds produced only in the oral portion of the vocal tract without the use of the vocal folds are called *unvoiced sounds*. The two nasal sounds, / *m* / and / *n* /, are both produced using the vocal folds, so they are voiced sounds.

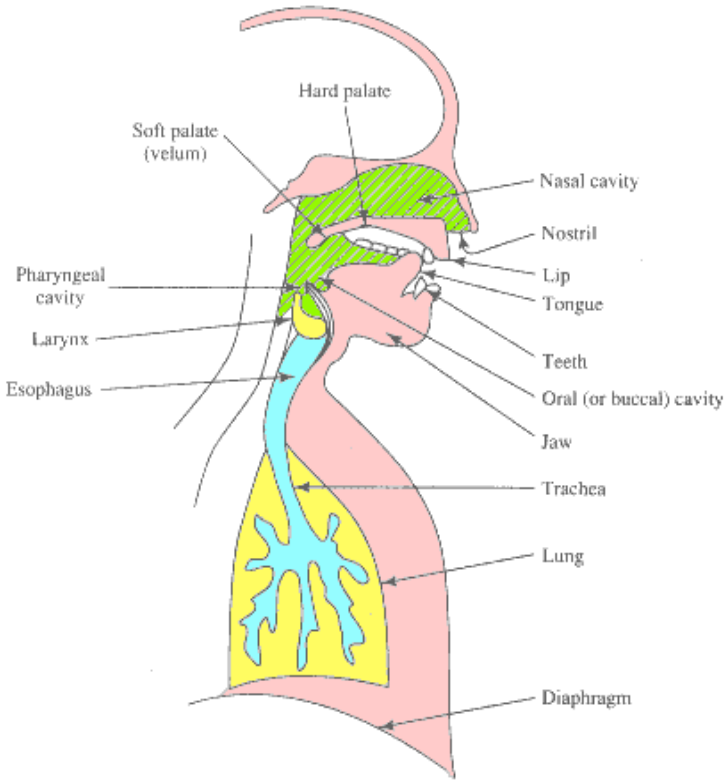

Figure 1.1
Schematic View of the Human Speech Production Mechanism, from which one can see the air going through Pharyngeal, Oral, and sometimes Nasal cavities, after being modulated by the vocal cords. Those three cavities act as three filters to produce different kinds of voiced sounds.

A source-filter model can be used to describe the speech production process of voiced sounds. In this model, the sound is produced at the vocal folds and is selectively modified or filtered by three cavities.

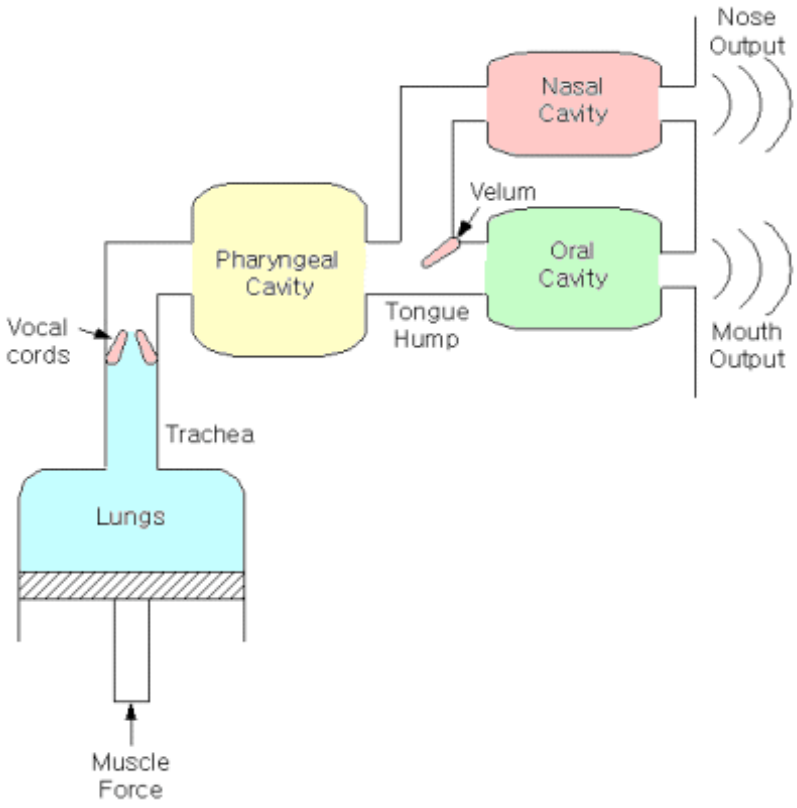

Figure 1.2
Block Diagram of Human Speech Production.

In the production of the vocal sounds, the vocal folds are drawn closely together by muscles, the air in the lungs is exhaled, the pressure below the vocal folds rises, and the closed folds are forced apart. The resulting rapid upward flow of air causes a decrease in pressure between the folds due to the *Bernoulli Effect* (explained later). The decrease in pressure, along with the elastic forces in the tissues, causes the folds to move together, partially blocking the passage and thus reducing the air velocity. The reduced air velocity increases the pressure below the folds and causes the process to repeat again. The sound produced in this manner is called a *glottal sound*.

The fundamental frequency of the resulting complex vibration depends on the mass and tension of the vocal folds. Men have longer and heavier vocal folds than women, with a typical fundamental frequency of about 125Hz; women are about one octave (a factor of two) higher, or 250Hz.

The glottal sound passes through several vocal cavities - the Pharyngeal (throat), oral, and nasal cavities, as mentioned above - that further change the sound of the wave emitted. The shape of the throat and nasal cavities is fixed for each individual and to a large extent determines the sound of the voice. They cannot be changed much voluntarily unless the speaker holds his nose and talks. The oral cavity changes shape through the movement of the tongue, lower jaw, soft palate, and cheeks to determine specific voiced sounds. The variation in shape of the nasal cavity for each speaker brings certain difficulties for the design of a speaker-independent algorithm.

It is necessary here to explain the meaning of *Bernoulli's Principle* in more detail. In a fluid flow situation, such as in water or air, the pressure in the moving fluid is lower at places where the speed of flow is greater. In the case of air rushing through the vocal folds, there is a lower-pressure area in the restricted region between the folds, leading to a Bernoulli force. This force and tension in the vocal folds causes the folds to close. Immediately after the vocal folds close, the air pressure builds up in the trachea, rapidly forcing the folds open once again. The burst of air through the vocal folds again creates the Bernoulli force, and the cycle is repeated. The rate of opening and closing determines the frequency of the resulting vocal sounds.

The frequency of vibration of the vocal folds is determined primarily by the controlled tension in the vocal folds, whereas the amplitude of the vibration is affected by increasing or decreasing the rate of the air flow between the folds.

1.2 Articulatory Phonetics for Nasals

Consonants and Vowels

The sounds of all languages fall into two major natural classes---*consonants and vowels*, often referred to by the cover symbols **C** and **V**. Consonantal sounds are produced with some restrictions or closure in the vocal tract as the air from the lungs is pushed through the glottis out the mouth. Different consonantal sounds are created when we change the shape of the

oral cavity by moving the articulators and changing the place of articulation in the oral cavity.

Nasal Speech Production

The symbols / *m* / and / *n* / represent the two consonants we discussed in this paper. Together with /□/ as in "thing", they are called *nasals*. When people produce / *m* / or / *n* /, air escapes not only through the mouth (when people open the lips), but also through the nose (nasal cavity). From Figure 1.1, the roof of the mouth is divided into the hard palate and the soft palate (*Velum*). Hanging down from the end of the velum is the *uvula*. When the velum is raised all the way and touches the back of the throat, the passage through the nose is cut off and air can only escape through the mouth (oral cavity). Sounds produced in this way are called *oral sounds*. For example, / *b* / as in "bed" is an oral sound. When the velum is lowered, air escapes through the nose as well as the mouth. Sounds produced in this way are called *nasal sounds*. The consonants / *m* /, / *n* / and /□/ are the only nasal sounds in English. All other consonant sounds are oral.

Common Properties of / *m* / and / *n* /

Besides belonging to the same nasal class, / *m* / and / *n* / also have other common properties. Both / *m* / and / *n* / are categorized as *consonatal voiced sounds*; both sounds are stopped completely in the oral cavity for a brief period of time, so they are *stops*, or *non-continuants*. Specifically, / *m* / is a *bilabial stop*, in which the airstream is stopped at the mouth by the complete closure of the lips; / *n* / is an *alveolar stop*, in which the airstream is stopped by the tongue making a complete closure at the alveolar ridge. Both are *sonorants*, instead of *obstruents*. (The obstruents cannot escape through the nose. They are either fully obstructed in its passage through the vocal tract, as in non-nasal stops and affricates, or partially obstructed as in the production of fricatives. The sonorants are produced with relatively free airflow either through the mouth or nose and thus have greater acoustic energy than their obstruent counterparts. Nasal stops are sonorant because the blocked air in the mouth continues to resonate and move through the nose.)

Phonetic Differences Between / *m* / and / *n* /

Phonetically, / *m* / and / *n* / have many similarities that cause difficulties when identifying them from each other. Nevertheless, they also have some phonetic differences when considering other classification criteria, such as place of articulation. / *m* /, together with / *p* / and / *b* /, are called *bilabials*, since they are articulated by bringing both lips together. / *n* / belongs to the *alveolar* class because it is articulated by raising the front part of the tongue to the Alveolar ridge. If speakers pronounce the word "new" (/ *nu* /), they can feel the tongue in close proximity to the bony tooth ridge as / *n* / is pronounced. The mouth is not closed and lips are open.

1.3 Acoustical Properties of Nasals

Acoustic characteristics of nasal consonants and nasalized vowels are perhaps the least well understood of all classes of speech sounds. One way to model nasal consonants is to assume that there are three tubes, or filters, one for each of the nasal, oral, and pharyngeal cavities. These filters modify the input signal generated from the vocal chords.

Before further discussion on this topic, some terminology is introduced.

1. Zeros and Poles of Filters

In the case of nasal sounds, the locations of zeros correspond to the frequencies of the anti-formants, and the locations of poles correspond to those of the formants [25]. We define zeros and poles before discussing anti-formants and formants. We assume the reader is familiar with linear time-invariant filters [30].

The transfer function of a linear time-invariant discrete-time filter is defined as

$$H(z) = \frac{Y(z)}{X(z)} \qquad (1.1)$$

where $X(z)$ denotes the z-transform of the input signal, and $Y(z)$ denotes the z-transform of the output signal.

If there exists one complex number z so that $Y(z) = 0$, we call z a *zero*. Similarly, if there exists a z so that $X(z) = 0$, we call z a *pole*.

The locations of zeros and poles provide useful insights into the performance of a filter. The locations are also important information when designing a digital filter.

For more details on zeros, poles, transfer functions, and other backgrounds of digital signal processing, refer to [28] and [30].

2. Formants

When examining a spectrum of a periodic signal waveform, we can see falls and rises in the spectral shape that span a wide frequency range (The unit is Hz in the case of human speech). These peaks, called *formants*, are estimates of the resonance of the vocal tracts. See Figure 1.3 for formants.

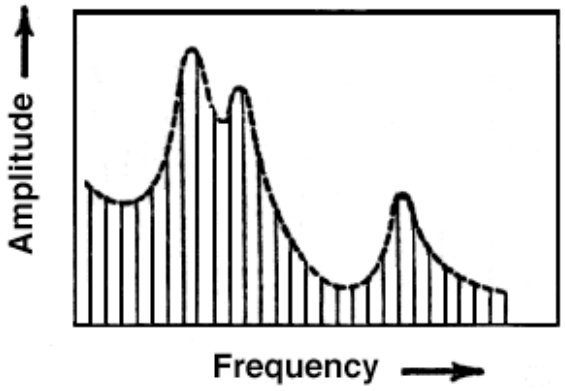

Figure 1.3
Spectrum of the Vowel "ah" Showing Three Formant Regions. The vertical lines represent harmonics produced by vibration of the vocal cords and based on fundamental frequencies. These harmonics are resonated by the vocal tract to create the vowel's characteristic spectral shape.

The locations of the formants depend on the size and shape of the filter (vocal tract). In the case of nasal consonants, the nasal and pharyngeal cavities constitute the filter. The shape

of this filter is approximately constant [25]. This means that if there is a change to the source but a minimal change to the filter, the formant center frequencies do not change and there is a minimal change in the long-term spectral trend of the combined sound output.

3. Anti-formant

The filter spectrum for nasal sounds is characterized by both formants and anti-formants. Anti-formants are the dips in the filter spectrum. They are introduced whenever there is more than one acoustic path from the source to the mouth opening. When producing nasal sounds, two acoustic paths are formed, from the glottis into the nasal cavity and from the glottis into the oral cavity.

4. Place of Articulation

Place of articulation is the relationship between the active and passive articulators (where to articulate, such as lips or tongue) as they shape or impede the air-stream. The active articulator usually moves in order to make the constriction. The passive articulator usually remains static and is approached by the active ones.

The International Phonetic Alphabet recognizes the following places of articulation in the table set forth below:

Table 1.1 Place of Articulation

Bilabial	The point of maximum constriction is made by the coming together of the two lips.
Labiodental	The lower lip articulates with the upper teeth.
Dental	The tip of the tongue articulates with the back or bottom of the top teeth.
Alveolar	The tip or the blade of the tongue articulates with the forward part of the alveolar ridge. A sound made with the tip of the tongue here is an apico-alveolar sound; one made with the blade, a lamino-alveolar.
Postalveolar	The tip or the blade of the tongue articulates with the *back* area of the alveolar ridge.
Palatal	The front of the tongue articulates with the domed part of the hard palate.
Velar	The back of the tongue articulates with the soft palate.
Uvular	The back of the tongue articulates with the very back of the soft palate, including the uvula.
Pharyngeal	The pharynx is constricted by the faucal pillars moving together (lateral compression) and, possibly, by the larynx being raised. "It is largely a sphincteric semi-closure of the oro-pharynx, and it can be learned by tickling the back of the throat, provoking retching"
Glottal	The vocal folds are brought together; in some cases, the function of the vocal folds can be part of articulation as well as phonation, as in the case of [h] in many languages.

Please see [4] for more detail.

Next, we consider the acoustical properties of nasals.

Zero vs Anti-formant / Pole vs Formant

Let $H(z)$ be the transfer function of the filter that creates the output sound from the input.

As mentioned earlier, in the case of nasal sounds, the locations of the poles correspond to the locations of the formants; the locations of the zeros correspond to anti-formant frequencies. The latter depends on the effective length of the oral cavity, and therefore on the place of articulation. Thus there are constant formants (because of the fixed shape of nasal cavity) and a movable zero (because of the variable shape of the oral cavity).

In the case of nasalized vowels, the mouth provides the main path and the nasal cavity is the shunt. Since the shape of the nasal cavity is fixed, the location of this zero is constant (typically at about 1500Hz). Hence instead of a movable zero being added to (and perturbing) a constant set of poles, there now is a fixed zero added to (and perturbing) the variable poles of the vocal tract [25].

From theoretical considerations of vocal tract modeling, it can be shown that the average spacing between formants is $C_{sound}/(2L)$ Hz, where C_{sound} is the speed of sound, $C_{sound} = 34000 centimeter/\sec ond$, and L is the total length of the vocal tract.

When producing nasal sounds, the formants depend on the combined nasal-pharyneal tract (with a length of about 20cm for an adult male), while a major anti-formant is introduced by the side-branching oral cavity (with a length of about 17cm for an adult male). Then, the spacing between the nasal formants is $C_{sound}/(2L)=$ 34000/40=850Hz. If the sampling rate is $F_s = 22050$Hz, the Nyquist limit of frequency is 22050/2=11025Hz. Hence, there exist 11025/850=13 formants in the 0-11025Hz range.

The single major anti-formant frequency depends on the length of the side-branching oral cavity, so its value could be varied for different kinds of places of articulation.

The above discussion shows that for a sampling rate of 22050Hz, the transfer function $H(z)$ should have 13 fixed poles and one single zero with a variable location when producing nasal sounds.

For more information on the relationship between zeros and anti-formants, and between poles and formants, please refer to Section 7.3 and 7.4 of [25].

Properties of Spectra of Nasal Consonants

The spectra of nasal consonants are characterized by nasal formants (labeled N1, N2, N3, …) which are due to the combined nasal-pharyngeal tube. For nasal and pharyngeal tubes with lengths typical of those of an adult male vocal tract, the first nasal formant, N1, is calculated to occur in the 300 to 400 Hz region; higher nasal formants occur approximately at 800Hz intervals [22][25].

In uvular and post-velar articulations, the oral cavity is effectively cut off from the nasal-pharyngeal tube and therefore has little effect on the resulting spectrum. For nasals produced with the tongue at the far back of the mouth, the spectrum is determined almost entirely by nasal formants. However, when the tongue articulation is further forward in the mouth, as in the production of palatal, alveolar, or bilabial nasals, the oral cavity acts as a side-branching resonator to the main nasal-pharyngeal tube and introduces oral anti-formants into the spectrum. The first anti-formant frequency occurs at a quarter-wavelength of the oral cavity, that is, at $C_{sound}/(4l_m)$ Hz, where l_m is the length of the side-branching oral cavity with a unit of centimeter, and C_{sound} is the speed of the sound. This implies that the frequency of the first anti-formant varies inversely with the length of the oral cavity, being lower for /*m*/ and higher for /*n*/.

In general, the effect of introducing oral anti-formants is to "flatten" the spectrum, particularly if nasal resonances and oral anti-formants coincide, and to lower its amplitude. In spectrograms, nasal consonants typically show overall amplitude dips and nasal formants that are very low in amplitude.

Moreover, oral formants occur in the spectrum when nasal consonants are produced. However, since these are likely to be close to the oral anti-formants in frequency, they are usually very low in amplitude [25].

Other Acoustic Characteristics of Nasal Consonants

Early studies have confirmed the following as acoustic characteristics of nasal consonants:

1. Nasal formants occur at approximately 700-800Hz intervals, beginning with N1 that locates around 250-300 Hz.
2. Nasal formant bandwidths are broader than those of oral vowels.
3. The first nasal formant has very high amplitude compared with that of higher formants.
4. Anti-formant exists in the 500-1000Hz range for / *m* / and in the 1000-2000 Hz range for / *n* /.

The last characteristic shows that the anti-formant regions of / *m* / and / *n* / are not overlapping. In fact, there are also some similar non-overlapping observations for the formant regions. It is agreed that the first three formants for / *m* / and / *ɲ*/ are 250, 1000, 2000Hz and 250, 2000, 2700Hz, respectively. So, theoretically, the spectrum should show a clear distinction on amplitude around 1000Hz between / *m* / and / *n* / cases. However, we cannot directly use the locations of formants and ant-formants to identify / *m* / and / *n* / for the following reasons:

1. They represent a set of harmonics with high energy, not just a single clearly identifiable harmonic, and within the band there is variation in terms of which harmonics, and with what energy, they contribute to the formant.
2. The shape of the formant/anti-formant is often variable over the course of a few dozen milliseconds.
3. The starting and ending points are co-articulation dependent.
4. There is a considerable amount of intra-speaker variation.

Place of Articulation and Nasal-Vowel Transition Boundary

There have been various studies that assessed the relative perceptual salience of formant transitions as cues for nasal place identification. The results of past studies have shown that the places of articulation cues are often guided by nasal-vowel transition regions in sound segments. In fact, the experiments of the last ten years, both in speech perception and in the

acoustic analysis of natural speech data, have suggested that the waveform of the nasal-vowel transition boundary is where the crucial information is contained.

More About Nasalization of Vowels

In the end, it is necessary to mention the concept of nasalization of vowels since in this thesis we will mainly focus on the transition regions between vowels and nasals. Nasalization of vowels is unavoidable during the production of those transition regions.

Vowels, like consonants, can be produced either with a raised velum that prevents the air from escaping through the nose, or with a lowered velum that permits air to pass through the nasal passage. When the nasal passage is blocked, oral vowels are produced; when the nasal passage is open, nasal or nasalized vowels are produced. In the English language, nasal vowels occur before nasal consonants in the same syllable. Oral vowels, however, occur before oral consonants.

Other rules, such as the feature-changing rule, also define the mechanization of nasals. The feature - changing rule ensures that / *m* / follows / *p* / or / *b* /, so the nasal and consonant match in places of articulation. If / *p* / and / *b* / can be identified and it can be determined that a nasal sound follows them, we can immediately conclude that this nasal sound is / *m* /, not / *n* /. This shows an indirect approach to identify the nasals in some special cases.

Chapter 2

Recent Progress on Distinguishing /m/ and /n/

In the past few years, researchers from the Voice I/O Lab of the Department of Computer Science at North Carolina State University have attempted to identify nasal consonants. They designed algorithms using the first two moments: mean and variance ([15][8]). In this chapter, those algorithms are briefly reviewed and their performance is re-evaluated. Although those algorithms do not provide convincing results by themselves, the idea of moment-space analysis does bring more information on the spectra of the nasals and will help design new approaches in the future.

As an important method on classification of nasal sounds, the Combined Spectra Method ([1]) is also summarized in this chapter. We will validate this method in the next chapter.

2.1 Moments Space Methods

In [15], moments of spectra, a measure of spectral shapes, are used to provide a direct mapping from the speech signal to parameters controlling the shape of the lips and position of the jaw during the articulation of the speech. The method requires no context, nor does it rely on any form of speech recognition. The two variables used to identify visemes or mouth shapes are the mean and the variance.

The following is a brief review of the algorithm for voiced sounds in [15].

1. Record the sounds at a sampling rate of $F_s = 22.050\text{KHz}$.
2. Identify a glottal pulse, GP, by a glottal pulse tracker [12]. The GP is related to the pitch by the following formula:

$$GP = F_s / Pitch \qquad (2.1)$$

where $Pitch$ is the pitch with unit Hz.

The tracker minimizes the sum of the first 4 odd harmonics of power spectra computed over increasing sample sizes in the beginning of $2GP$. The minimum occurs at $2GP$.

3. Average such harmonic of the power spectra of many sliding samples the same size as the GP to reduce noise. The spacing of the harmonics of the average is set to be the fundamental frequency, F_s / GP. The spectrum is clipped at 4KHz because, in voiced sounds, spectral moments (described in step 6) contain most information in this range.
4. Compute the cube root to deflate the influences of the first formant and interpolate to produce the spectrum, $S(freq)$.
5. Divide $S(freq)$ by the mass, M, to convert $S(freq)$ to a probability density function $p(freq)$.
6. Compute m_1 (first moment) and cm_2 (second central moment) from $p(freq)$.

The mathematical expressions for $S(freq)$, $p(freq)$, mean and variance are given by:

$$M = \int_0^{4000} S(freq)\,dfreq \tag{2.2}$$

$$p(freq) = S(freq) / M \tag{2.3}$$

$$m_1 = \int_0^{4000} freq \cdot p(freq)\,dfreq \tag{2.4}$$

$$cm_2 = \int_0^{4000} (freq - m_1)^2 \cdot p(freq)\,dfreq \tag{2.5}$$

The resulting moments have very little noise and are pitch independent.

This technique has proven itself for vowels and somewhat for *voiced and unvoiced fricatives* [15]. However, this approach does not work well to distinguish the places of articulation for English nasals in all contexts since /*m*/ and /*n*/ are acoustically similar.

The mean-variance pairs for all of the nasals for a single speaker lie in their own region of a 2-D moment space in which the horizontal axis is the mean and the vertical axis is the variance. It was shown that, as a group, the nasals are distinguishable from other English sound classes. However, the nasals themselves, in particular /*m*/ and /*n*/, are not distinguishable individually by this approach. The percentage of the overlap between /*m*/ and /*n*/ regions is very large so that no reasonable conclusion can be drawn.

The authors in [15] proposed to solve this problem by investigating the change in the shape of the normalized spectrum relative to the shape of the adjacent vowels to help identify the behavior of the anti-formants. It was claimed that, experimentally, for three speakers, over 90% of the cases were distinguished, with the main exception being the case when the signal transitions from the nasals to /*i*/. It was also claimed that, in most other cases, the path shape, together with its initial and/or final position in the 2-D space, have been sufficiently different to distinguish the two nasals, although the behavior can be radically different for each speaker. The most significant difference is the degree and direction of curvature. However, after performing more experiments, we found the above conclusions are not true. Therefore, this method does not perform well practically.

The authors in [8] described another approach to distinguish /*m*/ and /*n*/ by adjusting moments based on the velocity of the track moving into or out of the /*m*/ and /*n*/ region in moment space. It is suggested that the proper way is not to view the position of a few samples in moment space, but rather to examine how the track moves into the m-n region in moment space. However, this approach does not provide convincing results either when tested using more sounds.

The above discussion suggests that the mean-variance space approach cannot solve the identification problem for nasals on its own. However, this approach is based on the spectra of a single glottal pulse, and so it provides information on spectra around the transition part of the signal. In chapters 4 and 5, we will show how this approach is used to design new algorithms.

2.2 The Combined Spectra Method

It is well known that the acoustic relationship between the nasal and the vowel at the nasal-vowel boundary is highly informative for the /*m*/-/*n*/ distinction. In 1994, Harrington reassessed the contribution of *relational information* by classifying 1,946 syllable-initial, and 2,848 syllable-final, nasal consonants taken from continuous speech data of multiple speakers [1]. The relational information in the acoustic waveform is based on *difference spectra* and *combined spectra*, which are compared with *static spectra* (the spectra of pure nasals). Difference spectra are shown to perform more poorly than some kinds of static spectra. However, since classification scores from combined spectra are better than from either static or difference spectra, cues to nasal place of articulation can still be defined as relational. In the best scoring combined spectra, classification scores on *open tests* (in which different data is used in training and testing processes) are just under 94% correct for syllable-initial nasals and just under 82% correct for syllable-final nasals. These relatively high classification scores show that there is considerable information in the acoustic waveform for identifying nasal place of articulation from continuous speech data. In the next chapter, we will validate the Combined Spectra Method described in [1].

2.3 Other Progress

There has been very little progress in recent years regarding/*m*/ and /*n*/ identification. Presently, in [21], Berg and Stork comment that, “Sounds, such as '*n*' and '*m*', can also be easily analyzed because they are “long-lasting”. In both of these cases, the mouth end of the vocal tract remains closed, and the sound is therefore dominated by the formants of the nasal cavity. The nasal cavity acts like a Helmholtz resonator, with a formant at the resonant frequency of the resonator.” This statement can only be used to identify nasals as a whole class from other classes, and does not explain why nasals are "easily analyzed” by themselves simply because they are “long-lasting.”

Chapter 3

The Combined Spectra Method

From the discussions of Chapter 2, we can see that it is not easy to identify nasal sounds simply using the digital signal processing (DSP) method. We propose to supplement, using statistical classification methods. The DSP method extracts useful features. The statistical classification methods use those features as input to establish a classification model (or a classifier) using a training process. Finally the model helps to identify nasals using a testing process.

From a customer's view, when he initially uses the nasal classification software, he records his nasal sounds according to a given context. The recordings are used to establish the classification model through the training process. Next, the customer tests the model using an arbitrary speech context. The model should then accurately indicate which sound is / *m* /, and which is / *n* /.

There are several popular classifiers on this topic. The Combined Spectra Method is one of the most efficient. It uses spectra information as features and Principle Component Analysis (PCA) as the statistical classification method. For the past ten years, this method has been considered the best classifier on this topic because it has such a low error rate. This method is reviewed in this chapter.

3.1 Review of Ideas

In [1], experiments were made using a database of continuous Australian English speech produced by five male speakers (Their names were defined as: DB, DW, JC, MB, ND). All speakers produced a variety of Australian English that can be described as intermediate between General Australian and Cultivated Australian. The materials (1000 sentences, five passages) were recorded under excellent recording conditions, with sampling rate being 20 kHz.

In this chapter, the Combined Spectra Method is duplicated to see whether we can achieve the same high classification scores.

First, we need to locate the*release point*. Because the study is focused on the transition of nasals and vowels, it is important to locate the nasal-vowel boundary. This boundary is called the release point. The procedures for locating this boundary are the same as those described in [1]. The release point can be *visually identified* in the waveform as a break in the pattern of nasal pulses, and in the beginning of high-frequency components. Figure 3.1 shows an example of the waveform display of /*ma*/. The release point is labeled using the arrow in the middle of the figure. It is defined as the beginning of a pitch period that contains the first glottal pulse (called GP1) with incipient high-frequency energy. In addition to using visual inspection of the waveform display, *LPC (Linear Prediction Coding)* can help determine the release point. If there is a change in the pattern of spectra, particularly in comparison to the spectral pattern of the preceding nasal pulses, the current pulse is set to be the GP1 and its starting sample to be the release point. Finally, *perceptual testing* can also be used to double-check the results. In particular, the author listened to the recorded voices containing the vowel transitions to determine whether a nasal consonant was perceived. For more details on release point, refer to [2] and [3].

For most voice samples in the experiment, visual inspection and perceptual testing are enough to determine the release point. Moreover, the glottal pulse tracker described in [11] and [12] produces spikes wherever it goes through the release point, so it acts as an automatic way to locate release points.

The definitions of Hamming windows, such as *nasal-boundary* and *vowel-boundary* windows, are also the same as those described in [1]. However, instead of using a fixed length of 25.6 ms (512 samples) and 10 ms (200 samples) offset for every window, here the windows are three glottal pulses long with one glottal pulse offset. Using fixed lengths and offsets will result in considerable leakage when computing Fourier Transforms since the period can be slightly incorrect. The sounds are segmented manually so that they have exactly 3 glottal pulses. See Figure 3.1 for details.

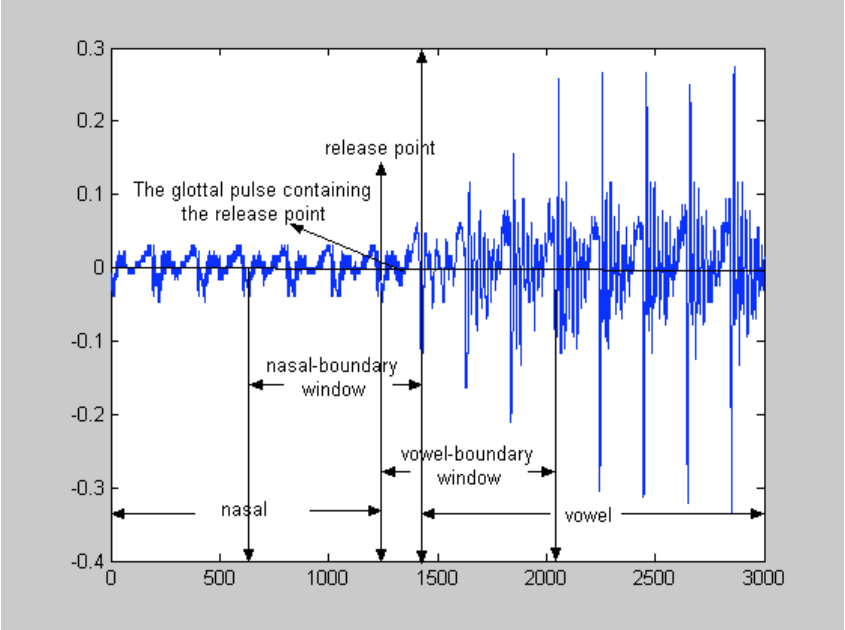

Figure 3.1
Release Point, the Nasal-boundary and the Vowel-boundary Windows

For more background information or details of implementation, please refer to [1].

1. Training and Testing

Before any statistical classification method can be implemented using the decision rules based on (Gaussian) probability densities, the means and covariance matrices for each class must be established. This is the *training phase* of the experiment. Having done this, the model can be used to classify speech samples, which is the *testing phase* of the experiment. Before the experiment, the researchers must know the class that each sound belongs to since the result of the classification need to be verified to determine whether the model performs well.

There are two different kinds of tests. In a *closed test*, the same speech samples are used in both the training and testing phases. In an *open test*, the two sets of samples are different. A closed set is valid only to the extent that the training sample is representative of the population as a whole, while an open test shows how well the model generalizes beyond the set of utterances used for training. Obviously, we will only focus on open tests. In a closed test, the classification accuracies should always be near 100%. See Chapter 9.4 of [25] for

more details.

For each speaker, there are 200 voice files. The files were segmented carefully. Part of the resulted sounds is taken to be the training sounds, and the rest to be the testing sounds. Notice that the training data is required to contain equal number of different kind of nasal-vowel combinations, such as /*am*/, /*im*/ and /*um*/.

2. Pre-emphasis

In 1971, Rosenberg did some perceptual experiments on glottal waveforms [25]. From his experiments, he found the spectrum falls at a rate of about 12 dB per doubling of frequency, or 12 dB/octave.

We also notice there are considerable losses to the acoustic energy in the vocal tract during speech sound production. One type of loss, energy loss, results when the acoustic energy radiates from the lips and nostrils in sound production. It causes a lowering of the resonance center frequencies and an increase in the bandwidths, especially in higher frequencies. Meanwhile, the loss produces one 6 dB/octave boost to the spectrum.

When the source spectrum is combined with the vocal tract (filter) to produce the spectrum of the sound, peaks occur at those harmonics that are closest to the formant frequencies of the vocal tract. The combination of the -12 dB trend caused by the source spectrum and the +6 dB boost produces a net downward sloping spectrum of -6 dB/octave.

In the spectral analysis of voiced speech, the -6 dB/octave trend is often compensated for by a pre-emphasis factor of +6 dB/octave to remove the downward trend. In this way, the intensity of high frequencies will not be very low because of the downward sloping spectrum.

The easiest way to do pre-emphasis is to subtract a scaled and delayed version of the signal from itself. The delay is one time-point and the scale-factor, a, is set to be just less

than 1. Typically, $0.96 \leq a \leq 0.99$. Thus, if the signal is $I(n)$, an approximate 6 dB/octave rise can be obtained from the filter (3.1)

$$O(n) = I(n) - aI(n-1). \qquad (3.1)$$

Since no pre-emphasis was done in [1], however, we omit this step in this thesis. Another reason is that we mainly use the information derived from lower frequencies instead of higher ones in the computation. More specifically, the highest frequencies we consider are only in bark 22, which is centered at 8500 Hz (See the following section). Hence we ignore pre-emphasis.

For more details on pre-emphasis, refer to Chapter 3 and 6 of [25].

3. Bark Frequency Scale

This scale was developed to capture the sensation of pitch differences in terms of "critical bands", which correspond linearly to length along the cochlea (1 critical band is equal to a distance of 1.3mm along the basilar membrane).

In this scale, equal distances correspond with perceptually equal distances so the bark scale represents the ability of the human ear to distinguish different tones at different frequencies. The use of the bark scale has the effect of stretching the vowel space where the human ear is most sensitive and contracting the space where tonal differences are difficult for the ear to perceive. The bark scale ranges from 1 to 24 Barks, corresponding to the first 24 critical bands of hearing. The published Bark band edges are given in Hertz as [0, 100, 200, 300, 400, 510, 630, 770, 920, 1080, 1270, 1480, 1720, 2000, 2320, 2700, 3150, 3700, 4400, 5300, 6400, 7700, 9500, 12000, 15500]. The published band centers in Hertz are [50, 150, 250, 350, 450, 570, 700, 840, 1000, 1170, 1370, 1600, 1850, 2150, 2500, 2900, 3400, 4000, 4800, 5800, 7000, 8500, 10500, 13500].

Figure 3.2 shows the relationship between Hertz frequencies and their bark equivalents, according to Traunmüller's approximation [97]. The crosses on the figure correspond to the

standard rounded bark scale. We can also notice that, above about 500 Hz, bark scale is similar to a logarithmic frequency axis; below 500 Hz, the scale becomes more and more linear.

The Traunmüller's approximation is defined as:

$$B = 26.81/(1 + (1960/ freq)) - 0.53 \qquad (3.2)$$

where B is in bark, $freq$ in Hertz.

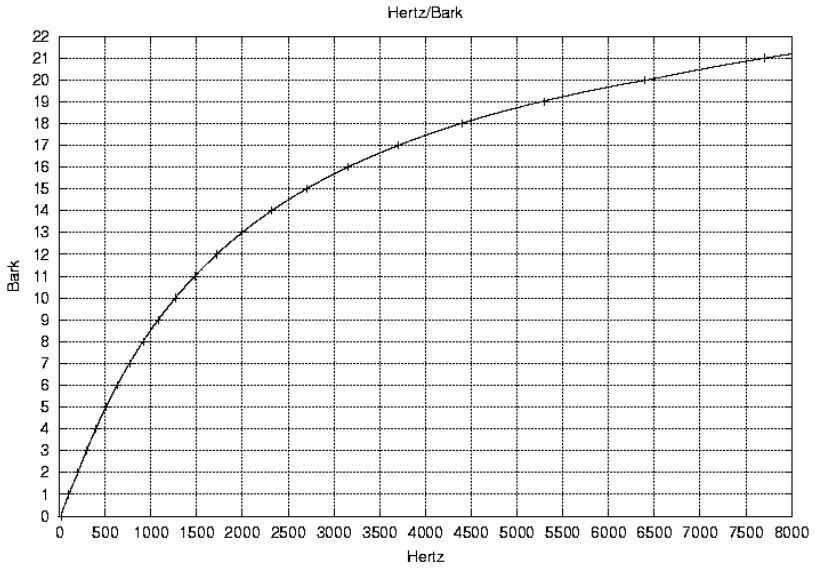

Figure 3.2

Bark Scale VS Frequency in Hertz

We will use one example to illustrate more clearly the relationship between bark scale and frequencies. Consider the following two figures. Figure 3.3 shows a spectrum; Figure 3.4 shows the bark scale counterpart of the spectrum. We can see that the shapes of the two representations are not alike at all. Now let's look at the process to transform a frequency to a bark scale. In Figure 3.3, when the coordinate of the horizontal axis takes 30 units, the magnitude is about 75. Notice the Nyquist frequency is 2050*(1/2) =11025 Hz, and there are 128 units totally in this range (See the horizontal axis in Figure 3.3). Then, every unit in the horizontal axis represents 11025/128 = 86 Hz. Now, 30*86 = 2584Hz, which is in bark 15

according to Figure 3.2. That explains the peak around bark 15. This peak is shown in Figure 3.4.

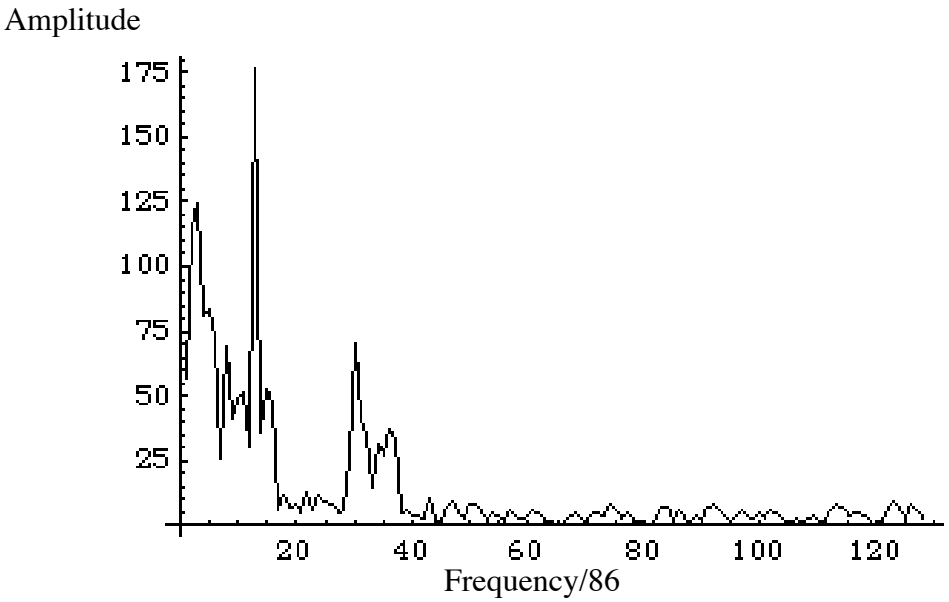

Figure 3.3
A Spectrum of Length 128 under a Sampling Rate of 22050Hz

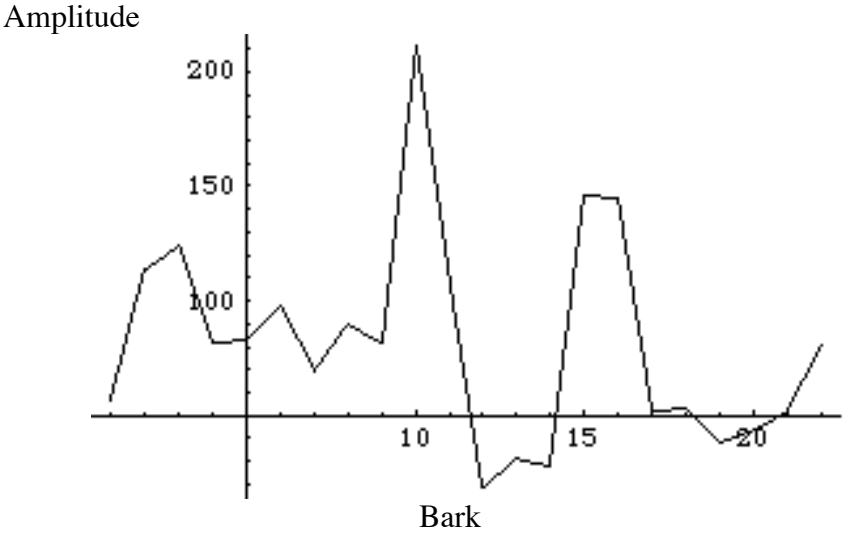

Figure 3.4
The 22 Bark-scale Representation of the Above Spectrum

4. Principle Component Analysis (PCA)

PCA is widely used in data analysis and dimensionality reduction. It is a data-reduction method that finds an alternative set of parameters for a set of utterances such that most of the variability in the data is compressed down to the first few parameters. The transformed dimensions in PCA are called *principal components*, and the new dimensions are guaranteed to be orthogonal and uncorrelated. Briefly speaking, for a zero-mean random vector x of dimension d, PCA tries to find k ($k \leq d$) orthonormal vectors so that the inner product of the random vector and the individual othonormal vector will have the largest variance. It can be shown that the k orthonormal vectors $\alpha_1, \alpha_2, ..., \alpha_k$ can be calculated by choosing the k eigenvectors corresponding to the k largest eigenvalues of $E(xx^T)$, which is the covariance matrix of x since $E(x) = 0$. The k orthonormal vectors form a basis of a subspace of R^d, and when x is projected to this subspace, it can be proved that the resulted random vector y is "closest" to x (the mean square error of $y - x$ is minimum) over the projection of x on any other subspace of R^d spanned by k orthonormal vectors. The PCA-derived vector $\beta \in R^k$ computed from $x \in R^d$ is referred to as the vector of k projection coefficients of x on the k eigenvectors $\alpha_1, \alpha_2, ..., \alpha_k$, so $\beta = [\alpha_1, \alpha_2, ..., \alpha_k]^T x$. This PCA-derived vector has components with the largest variances, so it can extract most of the randomness of the original vector.

For more theory and graph representations about PCA, refer to Chapter 9.6 of [25].

In the next section, we will talk about the implementations of PCA for the Combined Spectra method.

3.2 Algorithm, Result and Discussions

The following algorithm is based on Harrington [1]:

1. Convert the spectra of nasal–boundary windows and vowel–boundary windows into bark scale:

Calculate the spectra from the nasal-boundary windows and vowel boundary windows. Apply no pre-emphasis. Normalize the resulting spectral values obtained from each separate FFT by dividing them by the spectral value of the largest amplitude. Calculate the energy values in the first 22 critical bands (in bark scale) from each amplitude-normalized spectrum by summing all the spectrum values that fall within the separate critical bands.

2. Produce the combined spectra:

Create the combined spectra by concatenating 22 bark values from the nasal-boundary spectra and another 22 bark values from the vowel- boundary spectra into a single vector of length 44. Take these vectors of length 44 as the original *feature representations*. Let $\square_i, i = 1,...,n$ be n such feature vectors for training purpose. Calculate their mean vector, $fm = (fm_1, fm_2,..., fm_{44})$, and standard deviation, $fstd = (fstd_1, fstd_2,..., fstd_{44})$. Standardize the training data so that the resulted feature vectors, $\square'_i, i = 1,...,n$, have zero mean and unit standard deviation. Transform the testing data $\square$ into $\square'$ using the mean and standard deviation of the old training data, $\square_i, i = 1,...,n$, using the following formula:

$$\square'_i = \frac{\square_i \square fm_i}{fstd_i}, i = 1,...,44 \qquad (3.3)$$

where $\square = (\square_1, \square_2,..., \square_{44})$ is the original testing data; $\square' = (\square'_1, \square'_2,..., \square'_{44})$ is the standardized testing data.

Step 1 and 2 complete the process of Feature Extraction.

3. Implement PCA:

Training phase

Calculate the pooled covariance matrix C, which is defined as:

$$C = \frac{\sum_j (N_j - 1) C_j}{\sum_j (N_j - 1)}, \; j = 1,2 \qquad (3.4)$$

where C_j is the estimate of the covariance matrix of class j; N_j is the number of sounds of class j. Next, we calculate the corresponding eigenvalues. Discard a certain amount (actually, 44-k, where k is the number of the largest eigenvalues defined in Section 3.1.4) of the smallest eigenvalues and their corresponding eigenvectors. Regard the rest of the eigenvectors as the columns of a matrix V. $V = [\phi_1, \phi_2, ..., \phi_k]$. Here, $\phi_i, i = 1,...,k$ are the eigenvectors we keep and k is the number of such vectors. We know that k can be any number between 1 and 44. Then the PCA-derived feature vectors for training purpose can be calculated as follows:

$$\Phi^{PCA}_i = V^T \Phi'_i, i = 1,...,n \qquad (3.5)$$

where $\Phi^{PCA}_i, i = 1,...,n$, are the PCA-derived training feature vectors. Calculate the class centroids, $m_j^{centroid}, j = 1,2$, for each nasal class by taking the mean for the PCA-derived training feature vectors in this class. Note that the dimension of Φ'_i is 44, while that of Φ^{PCA}_i is k.

Testing phase

Transform the testing data Φ' one more time using the same matrix V derived from the training stage. Use the same transformation described in (3.5) to obtain PCA-derived testing feature vector Φ^{PCA}:

$$\Phi^{PCA} = V^T \Phi' \qquad (3.6)$$

Next, we will determine whether the sounds belong to class /m/ or class /n/ using the class centroids obtained in the training phase and PCA-derived vectors obtained in the testing phase.

4. Calculate Mahalanobis distance:

Assume our data are Gaussian distributed so that we can use the Mahalanobis distance measure [35].

The Mahalanobis distance r is defined as follows:

$$r^2 = (\vec{x}^{PCA} - m_j^{centroid})^T C_j^{-1} (\vec{x}^{PCA} - m_j^{centroid}) \quad (3.7)$$

where $\vec{x}^{PCA}$ is the PCA–derived testing feature vector obtained from (3.6), $m_j^{centroid}$ is the centroid for class j, and C_j is the covariance matrix for class j, $j = 1,2$.

5. Classify the testing sound:

We classify a testing sound the following way:

Calculate its Mahalanobis distance from its $\vec{x}^{PCA}$ to each of the two class centriods; find the smaller one; classify the sound to the class whose centroid is "nearer" to $\vec{x}^{PCA}$.

6. Apply experiments on different dimensions:

Repeat the above processes using different k, while k ranges from 1 to 44. Evaluate the performance for each case.

In our experiments, we segment the data files carefully so that the experimental objects have only 5 glottal pulses (two are nasal glottal pulses, two are vowel glottal pulses, and one contains the release point between the nasal and vowel). In the current stage, syllable-initial and syllable-final cases are tested separately so that the results can be more comparable to those in [1]. After combining the sound files obtained from all the five speakers, we have 342 sounds for /*m*/ and 389 sounds for /*n*/. For each nasal consonant, 150 sounds were taken as training data. Hence, there are 192 testing sounds for /*m*/ and 239 for /*n*/.

The result is shown in the following Figures:

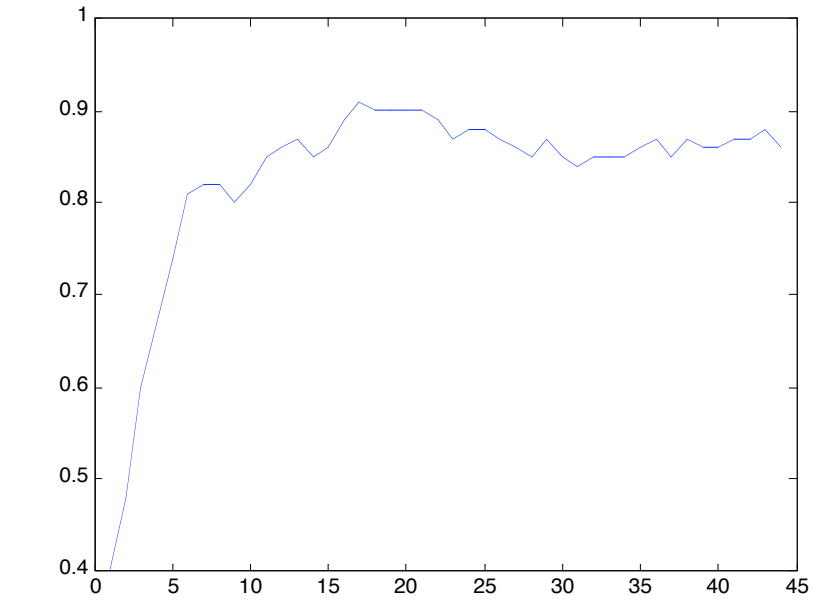

Figure 3.5
Syllable-initial Case

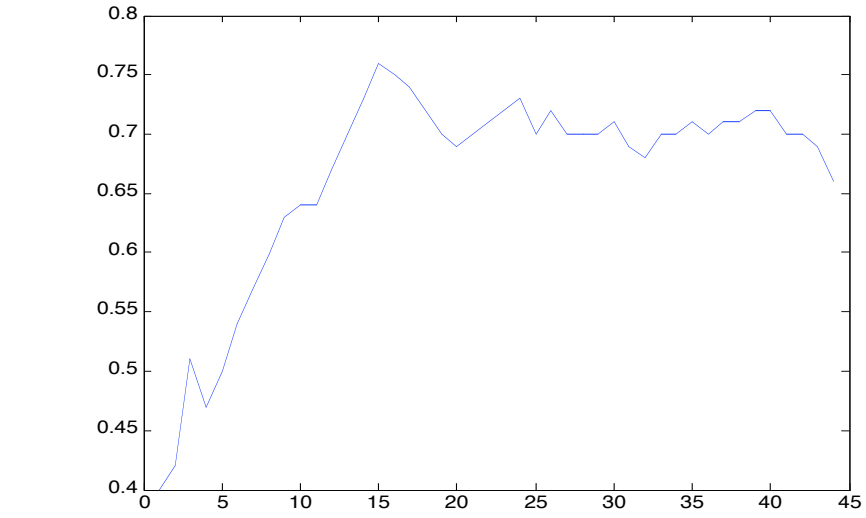

Figure 3.6
Syllable-final Case

In the above figures, the horizontal axis is k, the number of dimensions used in PCA; the vertical axis is the accuracy achieved when using the k eigenvectors corresponding to the k biggest eigenvalues with k between 1 and 44.

As seen, for syllable-initial case, when the first 17 eigenvectors are used, the classification score, 91%, is the best; whereas for syllable-final case, when the first 15 eigenvectors are used, the classification score, 76%, is the best. Those results are compatible with those of Figure 6 in Harrington [1]. We notice that the score in syllable-final case is not as good as that in [1], which are 94% and 82%, respectively. This is probably because our training data set is not as big as in [1] to produce a very accurate estimation of class representation.

We also notice that when too many eigenvectors are added, the classification scores for both cases become worse. There are two explanations:

1. When the number of eigenvectors increases, the dimension of the feature vector increases accordingly. While the training data set does not expand, the usefulness of each additional component is lessened. From a statistical perspective, the goodness of fit of a Gaussian model depends on having a large number of training sounds to estimate accurately the mean and covariance matrix. The model could be less accurate if the number of sounds is not increased while the number of components is increased since there could be more dependence between the components (See chapter 9 of [25]);

2. Variations from the new added directions might not provide more information on the identification; on the contrary, they may bring negative effects.

From this view, it is suggested that, given the fixed number of training data, we might want to do the following:

1. Calculate the correlations between the components and merge the highly correlated ones in order to ensure the independence of the rest;

2. Add new, independent and useful parameters into the model so that we can have more valuable information on the identification.

The validations of those two ideas involve discussions and theory supports on statistics, which will be discussed in Chapter 5.

In the next chapter, Chapter 4, we will talk about some DSP methods. Those methods extract reliable feature representations, which could be used as new parameters in the feature vector.

Chapter 4
Feature Extraction
Using
Singular Value and Eigenvalue Tests

In the Combined Spectra Method, the classification model was established using a feature vector of length 44, which is derived from the spectra around the transition regions of the nasal sounds. In this chapter, new feature parameters will be extracted which could be added into the feature vector as new components. We attempt to produce higher classification scores. The theoretical support and implementation of this idea is introduced in the next chapter. In this chapter, we discuss the extraction of the new features.

Our feature extraction approaches are based on singular-value and eigenvalue tests. Those approaches can successfully extract useful information on differences between nasals.

4.1 Singular Value Tests

Definitions

The Singular Value Decomposition (SVD) of a rectangular n by m matrix A is defined as

$$A = USW^T \tag{4.1}$$

where U is an n by n left orthogonal matrix, W is an m by m right orthogonal matrix and S is an n by m diagonal matrix of non-negative singular values. The columns of U form an orthonormal basis for the space spanned by the columns of A, while the columns of W form an orthonormal basis for the space spanned by the rows of A. The columns u_i and w_i of U and W are called the left and right singular vectors, respectively. The singular values, s_j, lie on the diagonal of S and ARE NORMALLY ARRANGED TO occur in descending order; the number of non-zero singular values represents the rank of the input matrix.

The properties of SVD are similar to those of the better-known eigenvalue decomposition. They both decompose the input matrix into a set of orthonormal basis matrices. SVD differs from the eigenvalue decomposition in that it is valid for any input matrix, while the eigenvalue decomposition is only defined for square matrices. Both SVD and the eigenvalue decomposition have been widely used to separate an input signal's spectrum into signal and noised components. SVD has one more advantage in that it can always give a real valued solution if the input matrix is real, whereas the eigenvalue decomposition may give a complex solution.

The SVD has a variety of applications in scientific computing, signal processing, automatic control, and many other areas. One important application on signal processing is Matrix Approximation:

By neglecting the small singular values in the "middle matrix" S in the SVD, we can obtain matrix approximations whose rank equals the number of remaining singular values. Since the singular values appear in decreasing order, the formula for the matrix approximation becomes

$$A_k = u_1 s_1 w_1^T + \ldots + u_k s_k w_k^T \qquad (4.2)$$

where k is the number of retained singular values. The terms $u_i s_i w_i^T$ are called the *principal images*. Often very accurate matrix approximations can be obtained with only a small fraction of the singular values.

Other applications of the SVD include computational tomography, image de-blurring, and geophysical inversion (seismology).

Application to Nasals

In this section, the SVD technique is used to try to distinguish nasals since the singular values are alternatives to frequencies, and it is expected that they can perhaps show something that the FFT does not.

Since the cues of place of articulation lie in the transition part between the vowels and nasals, the algorithm is implemented in the transition regions of the voice samples.

In Chapter 3, the experiments were conducted on a set of data recorded from different speakers. In this chapter, we use voice samples recorded from only one speaker since the automated system requires nasal identification of a single speaker.

After we locate the release point, as we did in Chapter 3, we take only five glottal pulses around it for our experiment to avoid the huge computation caused by calculation for singular values of an N by N matrix (defined next), where N is a large number (We will see later that usually N> 150). Those five glottal pulses are: GP1, two glottal pulses before GP1 and two after it. They are segmented carefully by hand so that we can estimate periods of the glottal pulses accurately. See Figure 4.1.

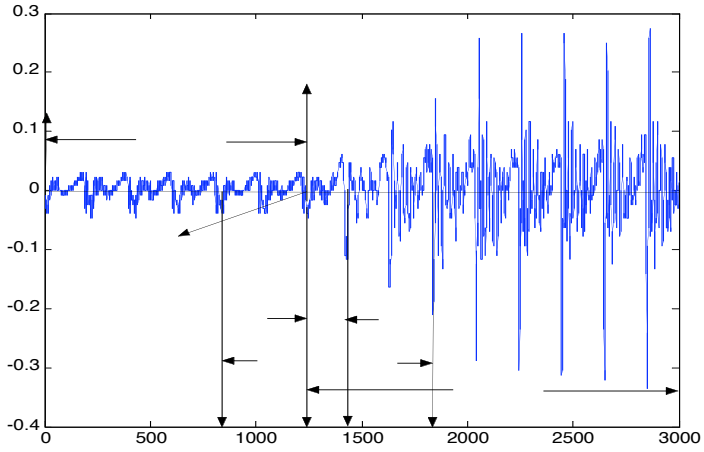

Figure 4.1
Release Point and the Five Glottal Pulses

If a glottal pulse contains approximately 200 samples, 5 glottal pulses is about 200*5/22050=0.045 seconds in duration where 22050 Hz is the sampling rate. Hence, we are experimenting on a very short nasal-vowel transition signal.

SVD algorithm I:

1. Read the values of the signal that contains only five glottal pulses; calculate its total length; divide this length by five, to get an estimation to the period of GP1; take this value as the window size.

2. Duplicate the last (the 5th) glottal pulse period several times and append them in the end of the signal (the five glottal pulses). This step ensures that when we start to shift from any sample in the 5th glottal pulse period, we have enough samples to generate the corresponding matrices (The matrix generation will be discussed next).

3. Start from the beginning of the signal, take the first N (=1/5*(signal length before appending in Step 2)) samples to form the first row of one N by N matrix.

4. Start from the second sample of the signal (that is, move to the right for one sample), and get another group of N samples sequentially. Those N samples form the second row of the matrix.

5. Continue moving by one sample at a time and get another group of N samples as a new row. When we form N rows, an N by N matrix is generated. Call the matrix M_1. Find the N singular values of M_1, and make those values the height values for the function $G(1,r)$ where r runs from 1 to N.

6. Now omit the first row of M_1 and add a new row in the bottom of M_1 so that the matrix is still square. The new row is formed by continually moving over one sample from the last row formed. Call the new matrix M_2. Make the singular values of M_2 the height values of the function $G(2,r)$ where r runs from 1 to N.

7. Continue this process until the original five glottal pulses are moved through. Until now we have generated a set of matrices, $M_i, i = 1,...,s$, where s is the total length of the five glottal pulse signal.

8. Calculate the singular values for every matrix and plot the function $G(i,r), i = 1,...,s$, and $r = 1,..., \mathrm{N}$.

9. Plot the largest (for example, 10) singular values of each matrix along the whole signal. Those singular values are much bigger in amplitude than the other ones and they may contain the information we need. Examine any changes in the nasal-vowel transition part from the 3-D graphs.

From the 3-D graphs of the singular values, we can see how those values change as the matrix moves over one sample at a time.

By examining the 3-D plots generated from the sounds containing the two different nasals, /*m*/ and /*n*/, it is expected that we can observe some differences on certain features, such as the frequency of change on magnitudes of the singular values. Hopefully, from those observations, uniform criteria can be drawn to identify /*m*/ and /*n*/.

Results

There are 8 graphs in the next 4 pages. In each page, thereare 2 graphs; the first one is the waveform in time domain of the signal and the second one is the 3-D graphs for the singular values.

In these 3-D graphs, axis 1 in the bottom left represents the number of the singular values; larger values are on the left-hand side, and smaller ones are on the right–hand side. In those figures, only the 10 biggest singular values are considered since the other ones have much lower amplitudes (See Figure 4.3b, for example). We assume that the smaller ones do not give as much information as the larger ones, and do not consider them. Axis 2 in the bottom right represents the number of matrices generated. The vertical axis 3 represents the amplitude of the singular values.

The first 2 pages contain graphs of one pair of syllable-final sounds *im* / *in* , while the next 2 pages contain those of one pair of syllable-initial sounds for the same vowel.

The graphs contain color representations, which have the following meaning:

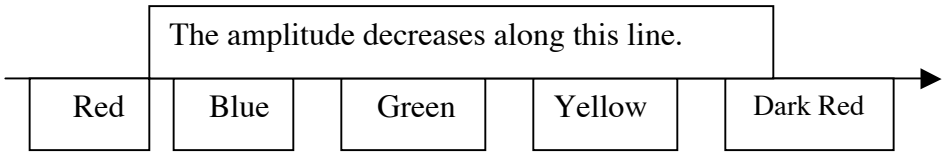

Figure 4.2
The Color and Amplitude

Sometimes, color makes the graphs too vague. It is not always the best way to show the differences in the transition area. However, the "mesh" function in Mathematica will make the graphs too dark since there are so many mesh lines to draw in every direction. Hence, in this study, we examine the graphs with the aid of color and animation.

It can be observed from the 3-D plots that the vowels have larger singular values than the nasals. Around the release point, in syllable-final cases, nasal sounds produce a sudden dip directly after the vowels diminish; in syllable-initial cases, as expected, vowels produce a sudden increase directly after the nasals diminish.

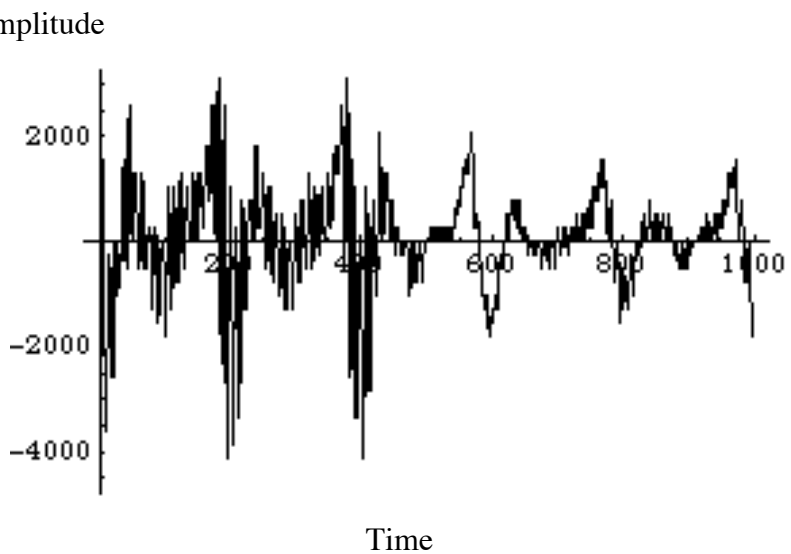

Figure 4.3a
Waveform of /im/

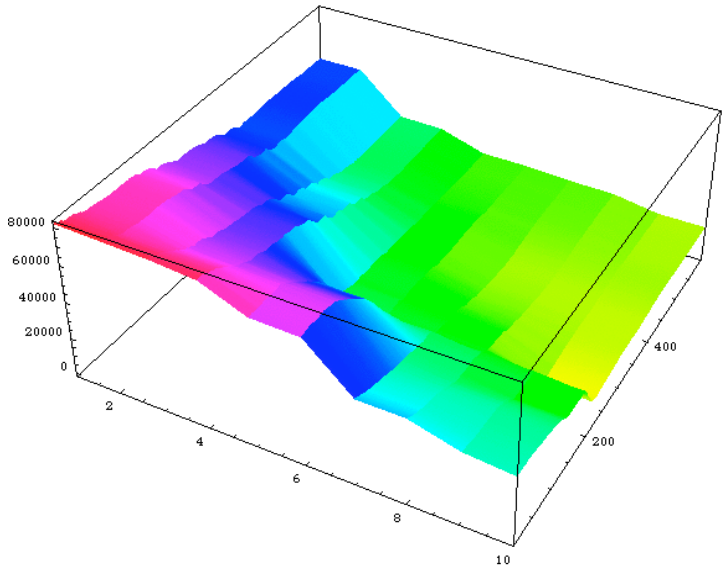

Figure 4.3b
The First 10 Singular Values Generated by the 3 Glottal Pulses in the Middle of the Signal

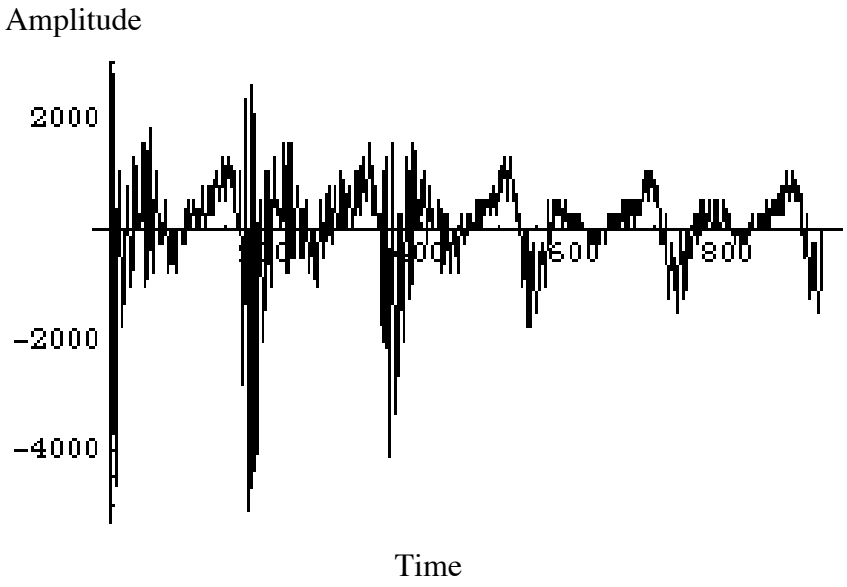

Figure 4.4a
Waveform of /in/

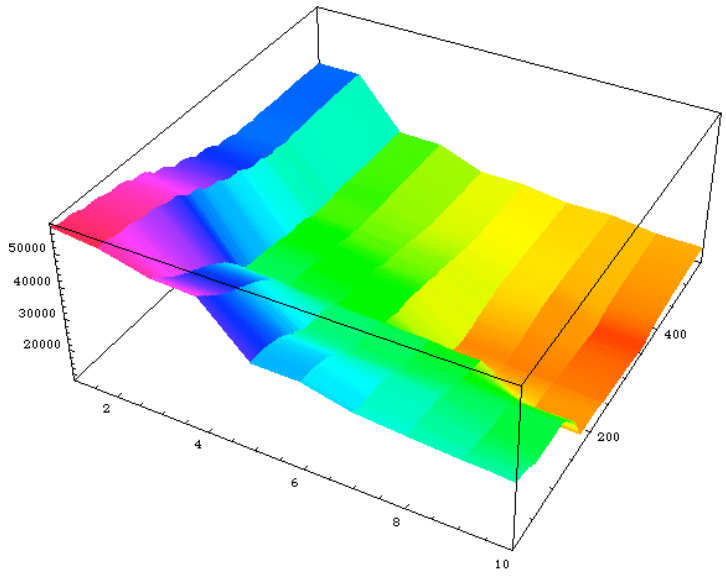

Figure 4.4b
The First 10 Singular Values Generated by the 3 Glottal Pulses in the Middle of the Signal

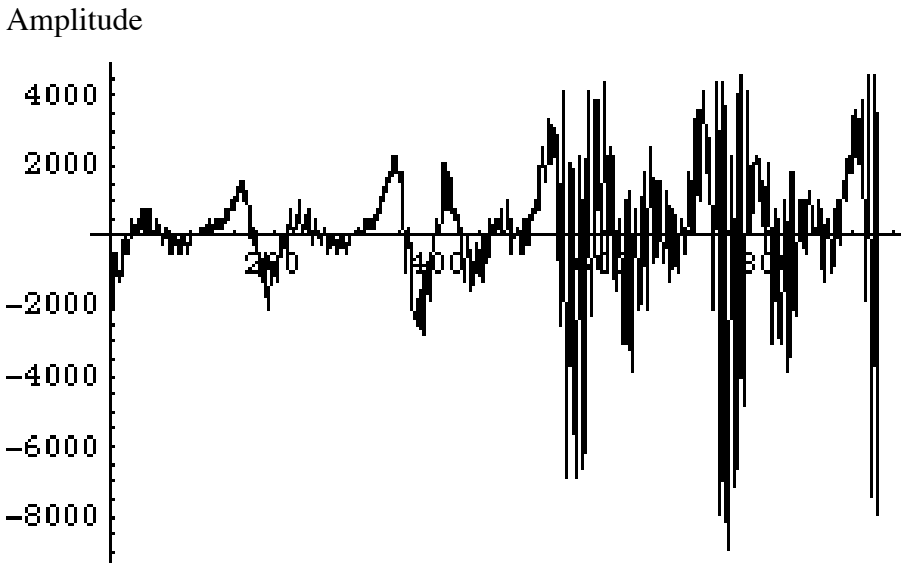

Figure 4.5a
Waveform of /mi/

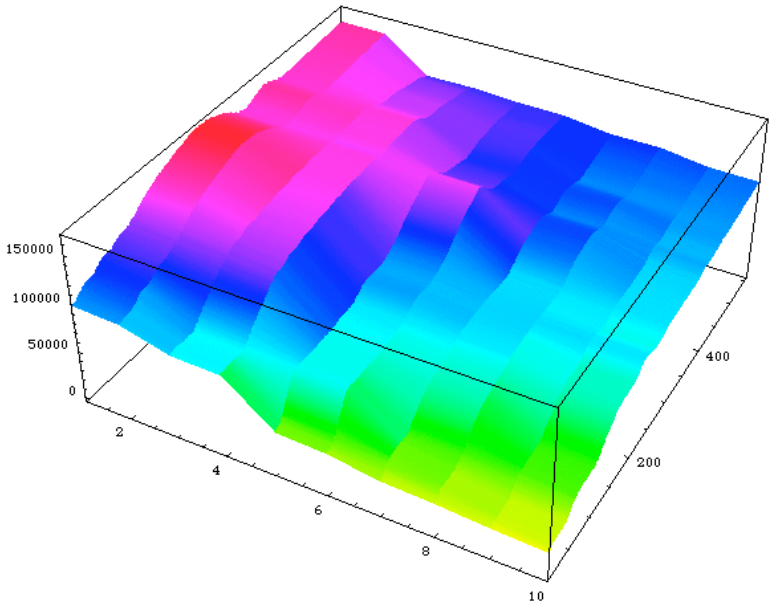

Figure 4.5b
The First 10 Singular Values Generated by the 3 Glottal Pulses in the Middle of the Signal

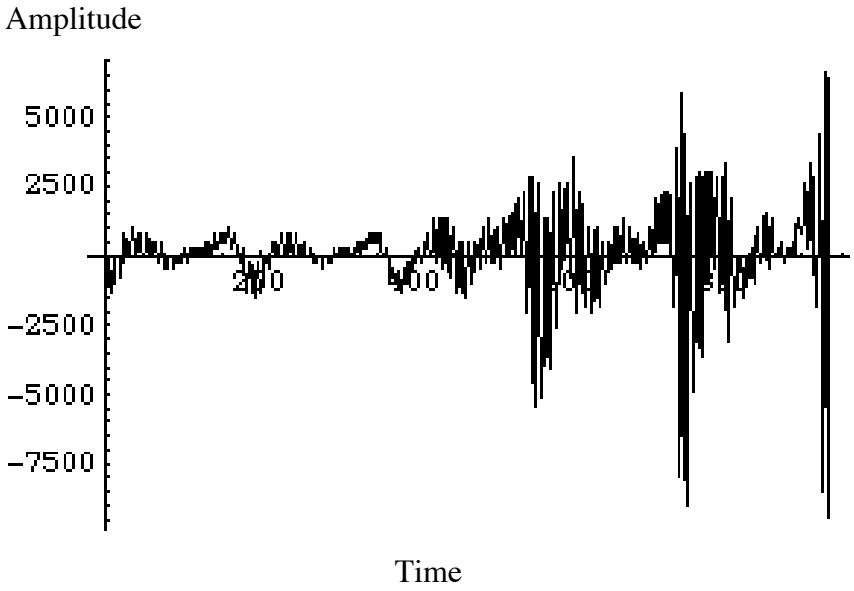

Figure 4.6a
Waveform of /ni/

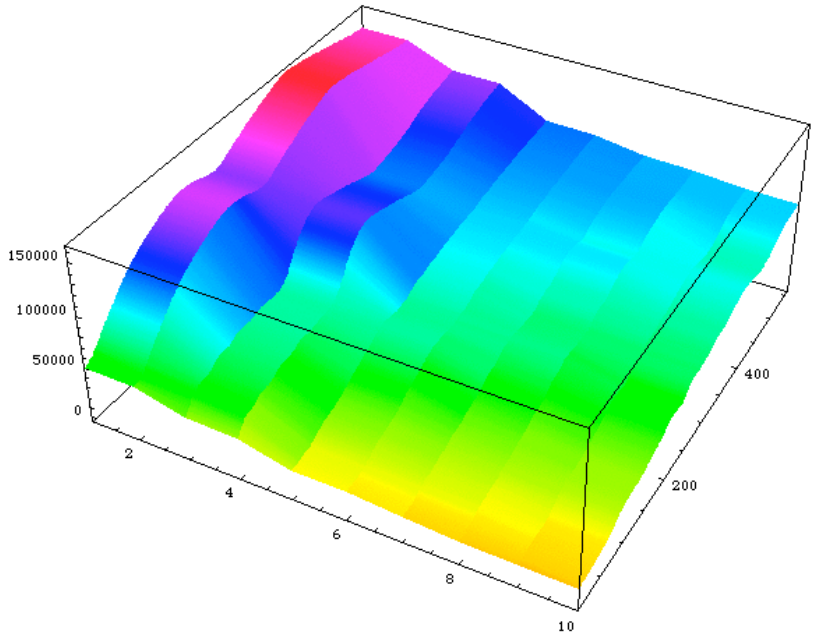

Figure 4.6b
The First 10 Singular Values Generated by the 3 Glottal Pulses in the Middle of the Signal

Here, it is necessary to discuss whether the singular values are stable along the pure nasal and vowel sounds. If they are not stable even in pure sounds, we need to find some way to smooth out the values so that we can use them for our analysis. For this question, there are several possible answers.

Stability

Suppose that we are dealing with a pure sound. Assume that all the glottal pulse periods have approximately the same number of samples. Let the period be N samples. We generate an N by N matrix M_1, using the same method described in SVD Algorithm I, and calculate the singular values. We move the starting sample to the next sample and generate another matrix, M_2. We know that rows 2, 3, …, N of M_1 are exactly the same as rows 1, 2, 3, …, N-1 of M_2, respectively. Since the period is approximately N for all the glottal pulses, we can notice that the values of row 1 of M_1 are approximately equal to those of row N of M_2. Thus, M_2 is approximately a permutation matrix of M_1.

To illustrate, let a signal be {GP1, GP2, GP3} = ((1.0, 2.0, 3.0), (1.1, 2.1, 3.1), (0.9,1.9, 2.9)), where each of the three glottal pulses has three samples. From the beginning, we get:

$$M_1=\{\{1,2,3\}, \{2,3,1.1\}, \{3,1.1, 2.1\}\}$$

and

$$M_2=\{\{2,3,1.1\}, \{3,1.1, 2.1\}, \{1.1, 2.1, 3.1\}\}$$

So, M_2 is approximately a matrix derived after doing three row permutations for M_1. Notice the matrix generated in this algorithm is always symmetric, which suggests that the eigenvalues of the matrix are real and they are equal to its singular values [19].

The following are some useful theoretical results:

Corollary 8.6.2 in [19]:

1. Permutations don't change singular values because they correspond to multiplications by orthogonal matrices.
2. Suppose the matrix A changes to $A + Error$: The ith singular value of A differs from the ith singular value of $A + Error$ by at most $\|Error\|$, where $\|\cdot\|$ is the Euclidean norm [19]. HOW DO WE COMPUTE THE EUDLIDEAN NORM?
3. If P is the circulant shift permutation that rotates the rows of A_1 up by one, we can then write $A_2 = P \cdot A_1 + Error$, where $Error$ is a matrix that is zero except for the last row. Then the ith singular value of A_2 differs from the ith singular value of A_1 by at most $\|Error\|$, where $\|\cdot\|$ is the Euclidean norm.

Hence, the N singular values of the sequential matrices will stay approximately the same. A signal of a pure nasal or vowel sound will have continuously stable singular values. See the following figures for singular values of a pure sound of /*m*/.

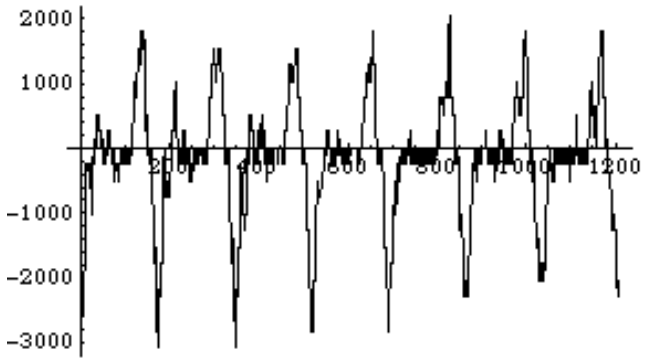

Figure 4.7
Seven Glottal Pulses for a Pure /*m*/ Sound in Time Domain

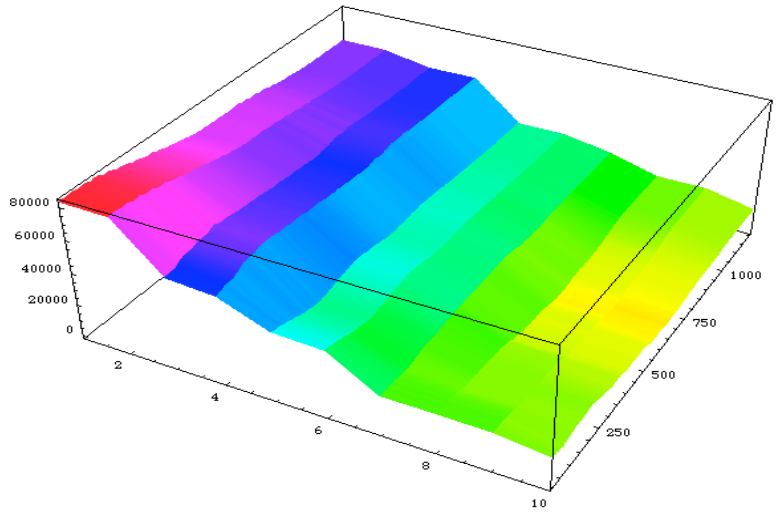

Figure 4.8
Plot of the Stable Singular Values along the 7 Glottal Pulses When the Size of the Matrix is Equal to the Period of the Glottal Pulse

However, when the size of the matrix is not approximately equal to the period of the glottal pulse, the stability does not remain. See the following figures.

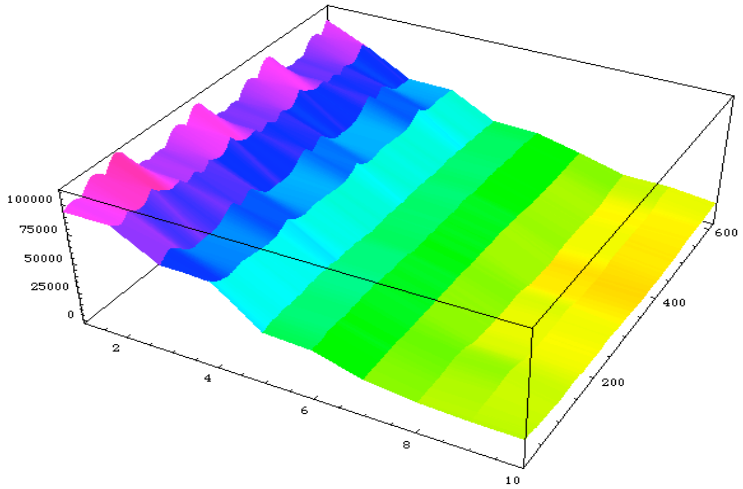

Figure 4.9
Plot of the Unstable Singular Values along the Same 7 Glottal Pulses When the Size of the Matrix is Equal to 1.2 Times the Period of the Glottal Pulse
Here only the singular values of the middle part signal (the middle three glottal pluses) are shown.

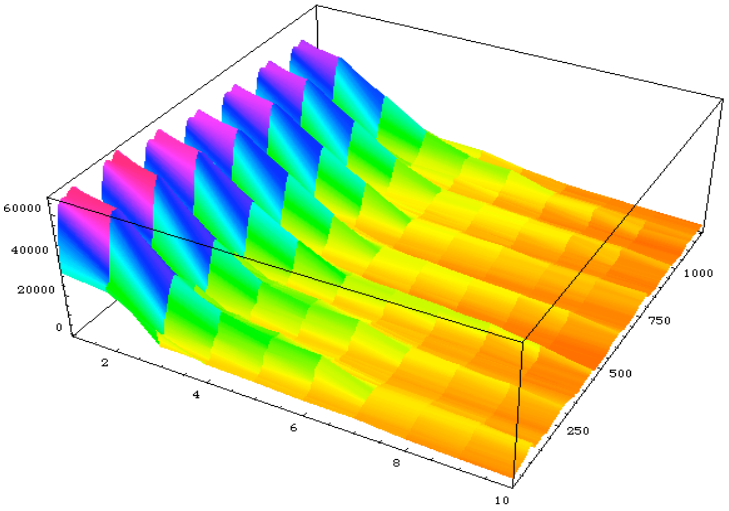

Figure 4.10
Plot of the Unstable Singular Values along the Same 7 Glottal Pulses When the Size of the Matrix is Half of the Period of the Glottal Pulse.
Here the singular values of the whole signal are shown.

To illustrate, let a periodic signal be

$$a = \{GP1, GP2, GP3, GP4\} = ((1,2,3,4), (1,2,3,4), (1,2,3,4), (1, 2, 3, 4)),$$

where the common period is 4 (samples), so P = 4.

Case I: N > P = 4

Without loss of generality, let N = 6.

Then,

$$M_1 = \{123412, 234123, 341234, 412341, 123412, 234123\},$$

$$M_2 = \{234123, 341234, 412341, 123412, 234123, 341234\},$$

$$M_3 = \{341234, 412341, 123412, 234123, 341234, 412341\},$$

$$M_4 = \{412341, 123412, 234123, 341234, 412341, 123412\},$$

$$M_5 = \{123412, 234123, 341234, 412341, 123412, 234123\},$$

$$M_6 = \{234123, 341234, 412341, 123412, 234123, 341234\}.$$

M_7={341234, 412341, 123412, 234123, 341234, 412341}

….

We can see that there are only P = 4 possible combinations of the rows: 123412, 234123, 341234, 412341. Every combination will occur at least once in every matrix. If the matrix has N>P rows, then there must be N-P (=2 in this case) repetitive combinations. For example, in M_1, 123412 and 234123 are repeated. The only 4 possible repetitive combinations are {123412, 234123}, {234123, 341234}, {341234, 412341}, {412341, 123412}. Notice they are not equal to each other. So only after P = 4 times' matrix generations, can we get two matrices which have the same rows (not necessarily in the same order). For example, M_1 and M_5, M_2 and M_6, …, have the same rows, and therefore have the same singular values. However, within P = 4 steps, no matrices have the same rows. So, the distribution of singular values has a period of P, which is also the period of the glottal pulses.

The above explains why, in Figure 4.9, there are 3 periods of singular values in 3 periods of glottal pulses.

Case II: N < P = 4

Similarly, we can prove that when N<P, the distribution of singular values also has a period of P too, which is also the period of the glottal pulses.

That explains why, in Figure 4.10, there are 7 periods of singular values in 7 periods of glottal pulses.

From the above, it is very necessary to keep the row/column size of the matrix the same as the period of the current glottal pulse when we use singular value methods. Since the stability would be conserved, we do not need to worry about the smoothing process, which may result in loss of information.

However, we can see from the graphs, when calculating singular values using this direct method, we do not obtain much useful observation to help identify nasals. The differences between / *m* / and / *n* / cases are not clear enough to create any classification criteria even after we normalize the singular values. Experiments have been done for multiple speakers, and each time we have the same observation.

SVD algorithm II:

In the following algorithm, instead of examining the singular values themselves, we use the singular values to generate the 2-D tracks in the mean-variance moment space, as we did using the frequencies in [5].

Assume the current GP contains N samples.

1. Calculate the singular values for an N by N matrix generated by the same way in algorithm I, starting from the first sample of the current glottal pulse.
2. Normalize the list of singular values by dividing it by the maximal value.
3. Interpolate with a cubic spline to produce a continuous singular value "Spectrum".
4. Convert to a pdf by dividing the "spectrum" by its mass, and then calculate m_1 (mean) and cm_2 (variance) in the range of 0 and N-1.
5. Repeat the above process until less than 3N samples remain.

This algorithm calculates only one single list of singular values for one glottal pulse; hence, we do not have to average "N" lists of singular values. No cube root is taken since there is no influence from the "first formant". The interpolation is done in the range of 1 to N. No scale is done for each moment.

See the following figures for results. Sounds are recorded from two speakers: Dave and Rodman. The horizontal axis represents the mean value, and the vertical axis represents the variance value.

Syllable-final cases:

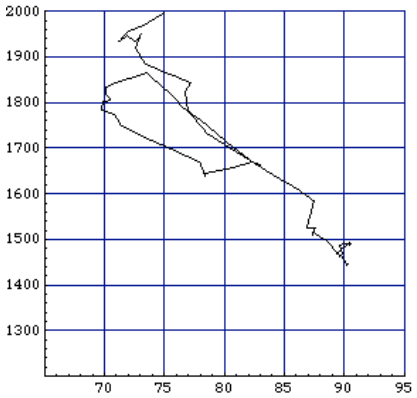

Figure 4.11a
Singular Value Track for /ahm/

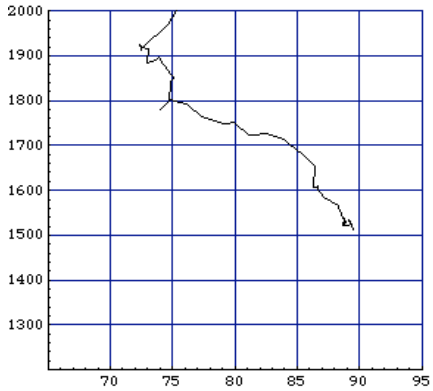

Figure 4.11b
Singular Value Track for /ahn/

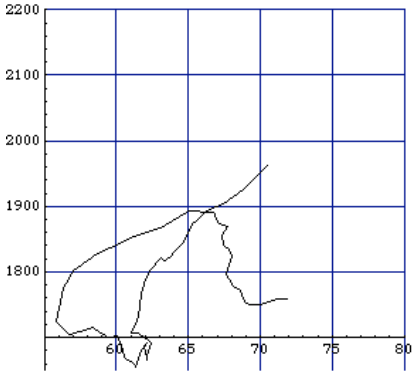

Figure 4.12a
Singular Value Track for /eem/

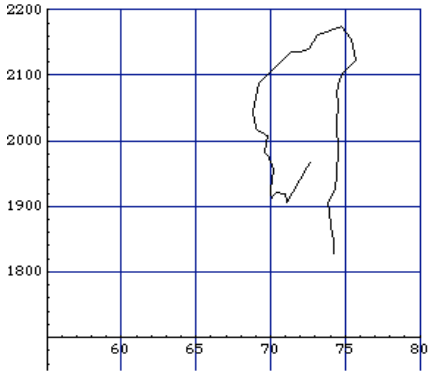

Figure 4.12b
Singular Value Track for /een/

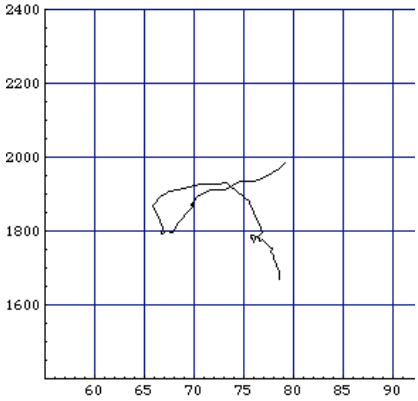

Figure 4.13a

Singular Value Track for /ehm/

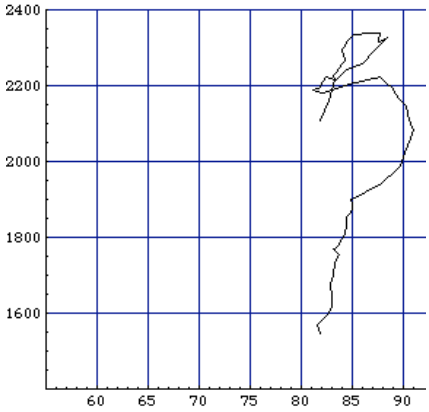

Figure 4.13b

Singular Value Track for /ehn/

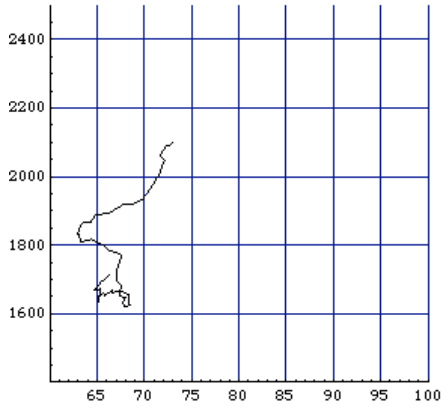

Figure 4.14a
Singular Value Track for /oom/

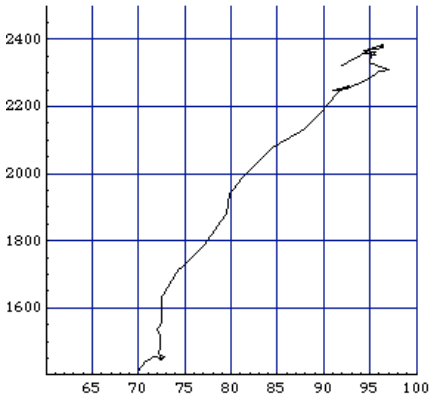

Figure 4.14b
Singular Value Track for /oon/

We can see, for the syllable-final cases, /*m* / tends to have an overlap in the track, while / *n* /, although showing curvature sometimes, tends not to have an overlap. In addition, except for the first case, the tracks for / *m* / and / *n* / occupy different regions in the moment space.

However, for the syllable-initial case, the result is not as clear as in the previous case.

Syllable-initial cases:

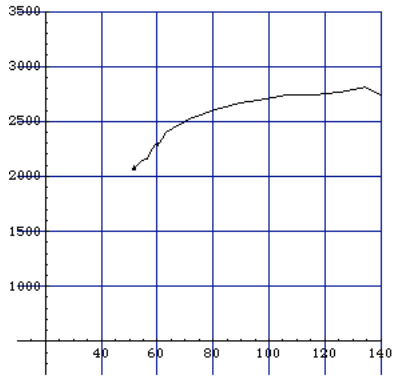

Figure 4.15a
Singular Value Track for /moo/

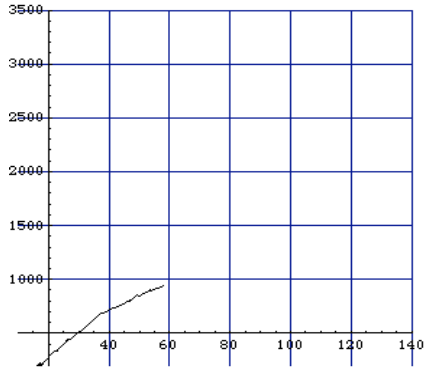

Figure 4.15b
Singular Value Track for /noo/

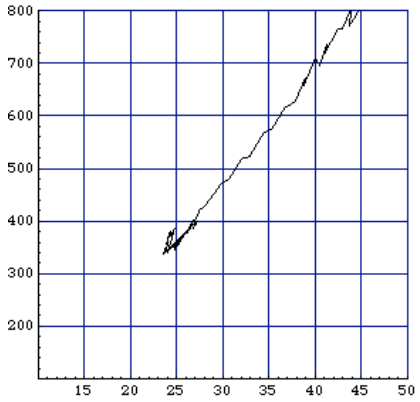

Figure 4.16a
Singular Value Track for /mee/

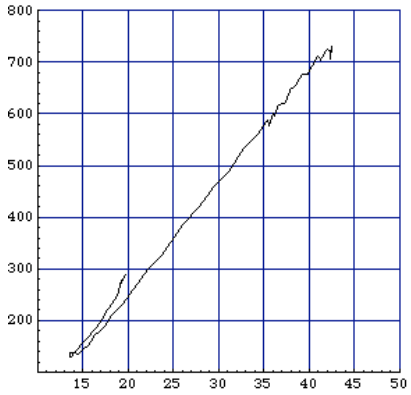

Figure 4.16b
Singular Value Track for /nee/

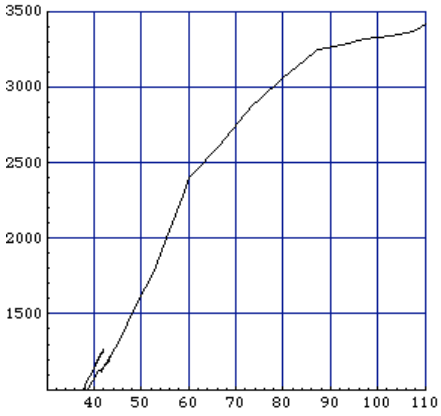

Figure 4.17a
Singular Value Track for /meh/

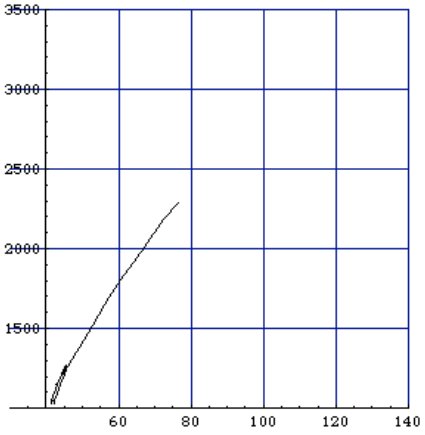

Figure 4.17b
Singular Value Track for /neh/

There is not much curvature for the tracks of both /*m*/ and /*n*/ sounds. All the tracks show more linearity. But, in most of the cases, the tracks of /*m*/ and /*n*/ cases tend to occupy different regions.

The moment values generated using this algorithm will be added into the classification model as new components of the feature vector since they provide useful information on the identification.

SVD algorithm III:

The difference between II and III is that algorithm III will produce N lists of singular values for each glottal pulse. Algorithm II will produce only one list of singular values for each glottal pulse. Thus, algorithm III is more like the one we presented in [5].

Assume the current GP contains N samples.

1. Calculate the singular values for an N by N matrix generated the same way in algorithm I, starting from the first sample of the current glottal pulse.
2. Shift over one sample.
3. Repeat step 1 and step 2 N times.
4. Average N lists of singular values and take the cube root, normalize it by dividing it by the maximal averaged singular value, interpolate it with cubic spline to produce a continuous singular value "Spectrum".
5. Convert to a pdf by dividing the "spectrum" by its mass, and then calculate m_1 (mean) and cm_2 (variance) in range of 0 and N-1.
6. Repeat the above process until less than 3N samples remain.

There is one major disadvantage for this algorithm: the cost of computation is very high. For each sample in the recorded sound, we need to form a matrix and calculate its corresponding singular-values. When we did experiments using a G4 Mac machine with memory 2GB and a processor speed of 1.4 GHz, it took about 20-30 minutes to run through a 1000 samples signal with period P being 200.

Due to its low efficiency, we consider SVD algorithm III to be future research, and therefore will not discuss it further in this thesis.

SVD of Non-square Matrices

Instead of using one N by N square matrix, it was suggested that we use one non-square matrix with a size of N by 20 to generate singular values. Without doubt, we can calculate the 20 singular values more quickly. However, the singular values we obtain will not be stable along the pure nasal sounds. Consequently, we need to design smoothing techniques to handle this problem, which might result in loss of information.

4.2 Eigenvalue Tests

Eigenvalue Decomposition (EVD) is widely used in speech signal processing. It is a crucial process of the Principle Component Analysis (PCA), the Independent Component Analysis (ICA), and the Periodic Component Analysis.

EVD algorithm I, II, III

EVD algorithms I, II and III are exactly the same as SVD algorithms I, II, III, respectively, except that calculating the singular values is replaced by calculating the absolute value of the eigenvalues. However, as mentioned in Section 4.1, for a symmetric matrix, the eigenvalues are the same as the singular values. Thus, the results are exactly the same as those in SVD algorithms.

EVD algorithm IV

This algorithm is a Metric-based algorithm.

From the analysis for both singular value and eigenvalue tests, we conclude that it is hard to distinguish any difference between / *m* / and / *n* / sounds directly from 3-D plots. It is desirable to design algorithms which can produce 2-D plots instead of 3-D plots.

From the above tests and the Voiced/Unvoiced Segmentor [7], one can see that, by shifting a window along the transition regions, we can examine the signal more carefully and then possibly extract useful information. EVD algorithm IV extracts spectrum information

using shifting windows and implements PCA to help identify nasal consonants. Instead of generating a singular value/eigenvalue vector of length N for each window, it is proposed to calculate a single (distance) value between two contiguous windows. From the distance values obtained when the algorithm goes through each sample of the signals, 2-D plots can be generated so that we can observe any feature distinctions between voice samples containing / *m* / and / *n* /.

The performance of a metric-based approach depends on both the distance measures used and the feature representation of the signals. The distance measure used in the Combined Spectra Method is the Mahalanobis distance. This distance is calculated in the following manner:

1. In the training phase, the centroids of classes / *m* / and / *n* / are computed.
2. In the testing phase, the Mahalanobis distances between the testing sounds and the two centroids are calculated.
3. Compare the two distances and attribute the sound to the class that is "nearer" to it.

However, in this section, classcentroids of / *m* / and / *n* / will not be calculated since we are using another version of Mahalanobis distance. For pairs of adjacent windows, we put all the windows on the left side into one category and all the ones on the right side into another category. New feature representations will be calculated from the original feature representations by projecting them into the two subspaces formed by the two above categories. The technique to form the subspaces is PCA. Next, we calculate the distance r between the two new feature representations (PCA-derived feature vectors).

Since it is often complicated to directly compute the dissimilarity between two collections of vectors, both vectors are often individually modeled parametrically as single or multiple Gaussian distributions. Then distance measures between the two parametric statistical models can be applied [35].

The Mahalanobis distance used in this section has a different form from equation (3.7):

$$r^2 = \frac{1}{2}(\mu_1 - \mu_2)^T (C_1 C_2)^{-1} (\mu_1 - \mu_2) \qquad (4.3)$$

Here, the assumption is that the two sets of vectors are both multivariate Gaussian distributed with means being $\mu_i, i = 1,2$, and covariance matrices being $C_i, i = 1,2$, respectively [35].

A high distance value indicates where a possible acoustic change occurs, and it is expected that / *m* / and / *n* / will produce certain differences from this feature representation.

The speech signals were recorded with a sampling rate of 22050 Hz, and 8 bit quantization. The feature representation used is bark scaled spectrum.

Implementation

Notice the PCA does not use the singular values, but uses theeigenvalues. That is why we regard this approach as one eigenvalue test.

To perform the experiments, we use the same method as in SVD to preprocess the sound files: manually pick up 5 consecutive glottal pulses so that the middle glottal pulse contains the release point.

The algorithm is described as follows:

1. Let $win1 = 32$ samples, $win2 = 2 \cdot win1 = 64$ samples; $shift = 4$ samples;
2. Calculate, t, the number of iterations using the following formula:

$$t = \frac{Length(signal) - win2}{shift} + 1 \qquad (4.4)$$

This is because we need to shift a left and a right window at one time; each pair of two windows has a length of 64. See Figure 4.25;

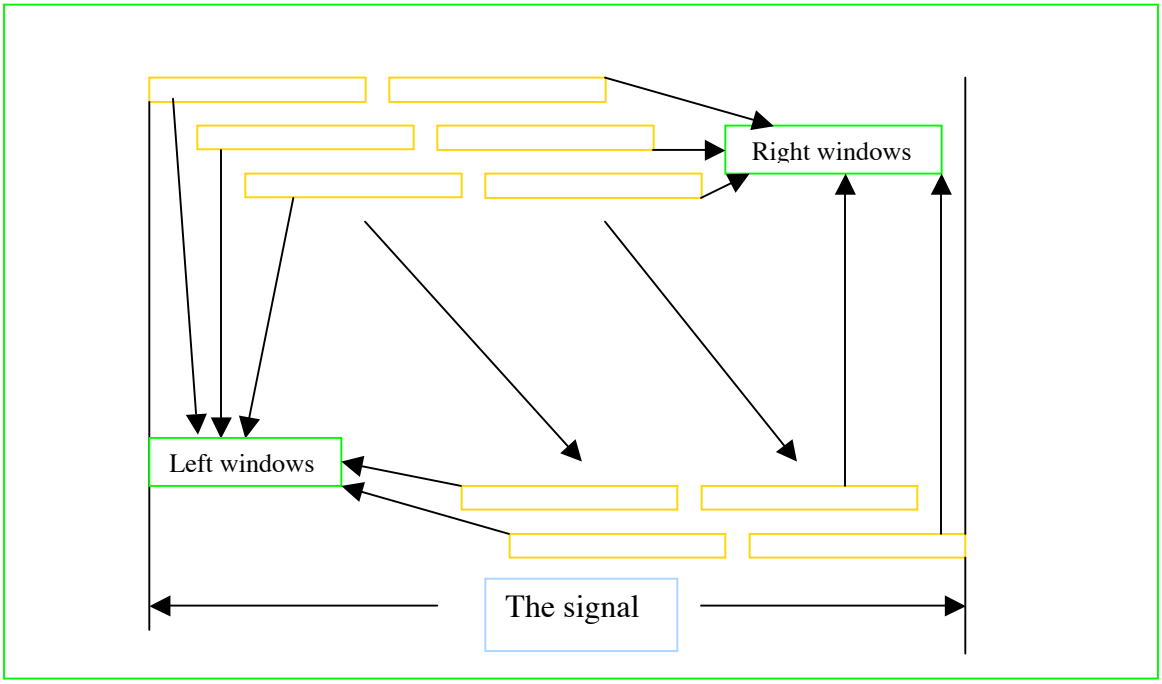

Figure 4.18
Left and Right Windows

3. Use Hamming window of length 32 and shift value of 4 to obtain the spectrum for the left and the right windows;
4. For each window, transform the spectrum from frequency scale to bark scale, take the first 22 bark values as the original feature representation, process the feature vectors of the left windows so they have zero mean, and do the same for the right windows;
5. Perform PCA to map the original features of each window onto the corresponding eigenspaces in order to form the PCA-derived features. In this approach, two eigenspaces are constructed separately, one for the left windows and another one for the right windows. To state mathematically, consider the two windows of feature vectors, $Q^L = \{q_i^L\}_{i=1,\ldots,n}$, and $Q^R = \{q_i^R\}_{i=1,\ldots,n}$. Assume q_i^L and q_i^R are samples of two zero mean Gaussian distributed vectors Q^L and Q^R respectively. Then the covariance matrices of Q^L and Q^R can be calculated respectively. Let $V^L = [\square_1^L, \square_2^L, \ldots, \square_k^L]$, whose columns are the eigenvectors corresponding to the

largest k eigenvalues of the covariance matrix for Q^L. Similarly, let $V^R = [\square_1^R, \square_2^R, ..., \square_k^R]$, whose columns are the eigenvectors of the largest k eigenvalues of the covariance matrix for Q^R. Then the PCA-derived features of the two windows can be calculated from:

$$Q^{PCA,L} = \{q^{PCA,L}{}_i = V^L (V^L)^T q_i^L\}_{i=1,...,n}$$

and

$$Q^{PCA,R} = \{q^{PCA,R}{}_i = V^R (V^R)^T q_i^R\}_{i=1,...,n}$$

respectively.

6. Calculate the distances between the PCA-derived features for each left and right window pair, plot the result, which is a 2-D graph.

In order to avoid any confusion, I am using an example to illustrate the process.

Assume the signal is s = {1, 2, 3,,4.., 24}, let $win1 = 8$, $win2 = 2 \cdot 8 = 16$, and the shift value is 2. The iteration number is (24-16)/(2) +1= 5.

Then:

win1_left is {1, 2, 3, …, 8}, win1_right is {9, …, 16};

win2_left is {3, 4, 5, …, 10}, win2_right is {11, …, 18};

win3_left is {5, 6, 7, …, 12}, win3_right is {13, …, 20};

win4_left is {7,8,9, …, 14}, win4_right is {15, …, 22};

win5_left is {9, …, 16}, win5_right is {17, …, 24}.

After the bark scale spectrums are calculated for the above windows, every window has a corresponding feature vector with length being 22. Assume the windows are labeled as $F_j^i, i = L, R, j = 1,...,5$. Then F_1^L, F_2^L, F_3^L, F_4^L, F_5^L are samples of Q^L, and F_1^R, F_2^R, F_3^R, F_4^R, F_5^R are samples of Q^R. Process all the vectors in Q^L so that they have zero mean; do the same thing for Q^R. Calculate the largest k eigenvalues of the covariance matrices for Q^L and Q^R, respectively, and get V^L and V^R; compute PCA-derived features for each window using V^L, V^R and the bark scaled features; calculate the distance between PCA-derived features of adjacent windows; plot the distance in a 2-D space.

Results

It is desirable to find some feature distinctions between the 2-D plots of $/m/$ and $/n/$.

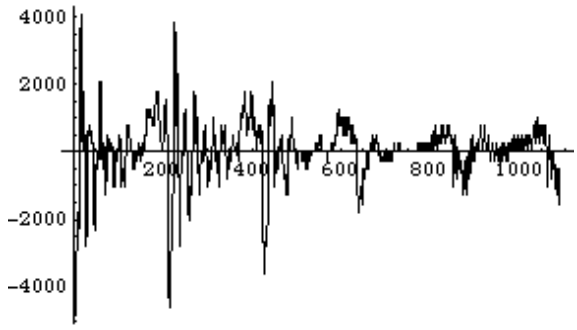

Figure 4.19a
Waveform of /am/

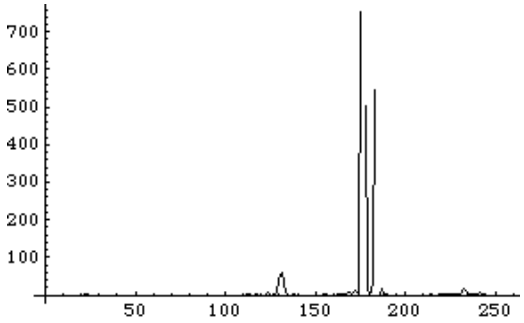

Figure 4.19b
The Spike Generated by EVD IV

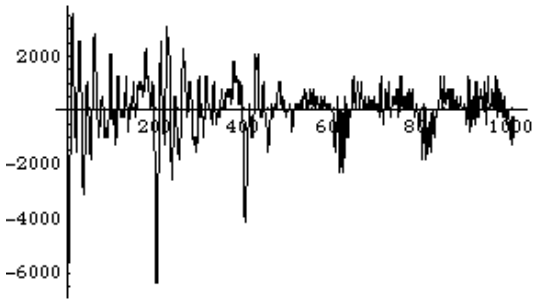

Figure 4.20a
Waveform of /an/

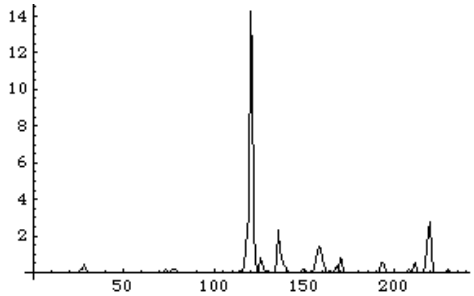

Figure 4.20b
The Spike Generated by EVD IV

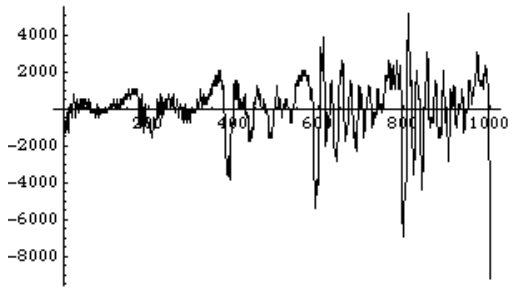

Figure 4.21a
Waveform of /ma/

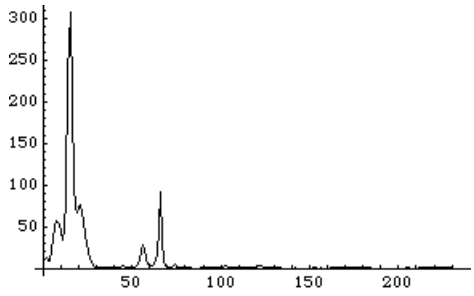

Figure 4.21b
The Spike Generated by EVD IV

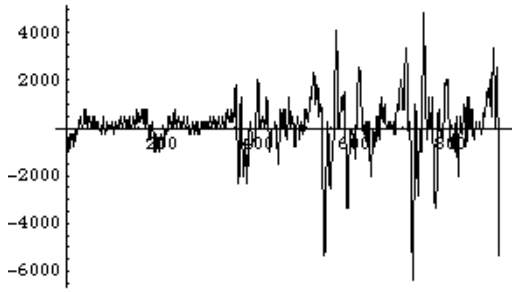

Figure 4.22a
Waveform of /na/

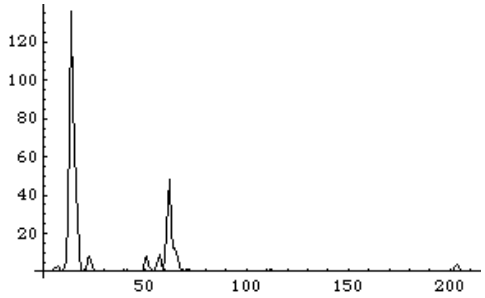

Figure 4.22b
The Spike Generated by EVD IV

From the above figures, we can see for a single speaker, in the syllable-final case, there are spikes in the transition region of the signal and the amplitude of / *m* / sounds is larger than that of / *n* / sounds. In the syllable-initial case, the observation is similar. This is true for 18 out of 20 pairs of sounds of 3 speakers.

However, if the signal is initially normalized by dividing the maximal value, then in the syllable-initial case, the amplitude of / *m* / sounds is always less than that of / *n* / sounds. This is an interesting observation.

From the above figures, we can see that the graphs have very similar shapes. Most graphs share the same pattern, which seems to be several spikes on a rather smooth surface of distances. One spike is always shown in the beginning (syllable-final case) or in the end (syllable-initial case) of the nasals. This is another interesting observation, although it is not about the difference between the nasals. Inspired by this by-product of this method, we can combine this method and the glottal pulse tracker to accurately locate the release point automatically. This, however, is a topic for future research.

4.3 Conclusions

From the above discussions, we conclude that there are indeed cues to identify nasals if we examine the transition regions using singular value or eigenvalue tests. SVD algorithm II and EVD algorithm IV can extract some useful information on nasal identification. It is suggested that the subtle differences between nasal sounds can be accumulated or enlarged if we make appropriate use of this information.

In the next chapter, we will use the information we have to produce more efficient feature vectors, and then process them using statistical classification techniques. We will compare this performance with the Combined Spectra Method.

Chapter 5

Statistical Classification Techniques

In this chapter, we apply statistical classification techniques and pattern matching methods on a certain feature parameter set to identify /*m*/ from /*n*/ for one single speaker, possibly with a small number of training samples.

Before implementation of any statistical techniques, we consider factors such as adding more useful components in the feature parameter set and dimensionality reduction in order to design a model (with less dimensions) that only uses a small number of training samples in the training phase.

Assumptions and implementations of several statistical classification techniques are then introduced, and experimental results are summarized. The techniques discussed in this chapter include PCA, Linear and Quadratic Discriminant Analysis (LDA and QDA), and Support Vector Machine (SVM).

We begin by introducing the traditional Bayes Rule and other statistical quantities.

5.1 Bayes Rule and Discriminant Functions

Bayes Decision Theory

Assume there is some *a priori* probability $P(i = 1)$ that the next observation is /*m*/, and some *a priori* probability $P(i = 2)$ that it is /*n*/. The *a priori* probabilities reflect the priori knowledge of how likely the next observation is /*m*/ or /*n*/. Obviously, $P(i = 1) + P(i = 2) = 1$ and $P(i = 1)$, $P(i = 2) \geq 0$. Usually, we estimate $P(i = 1)$ and $P(i = 2)$ simply by empirical frequencies of the training set:

$$P(i=1)=\frac{N_1}{N} \text{ and } P(i=2)=\frac{N_2}{N} \tag{5.1}$$

where N is the total number of training samples, N_1 and N_2 are the numbers of samples in $/m/$ and $/n/$, respectively.

If a feature vector, say, x, is available, we can define $p(x \mid i=1)$ and $p(x \mid i=2)$ to be the conditional PDF (probability density function) for x, given that the next observation is $/m/$ and $/n/$, respectively.

Suppose we know $P(i=1)$, $P(i=2)$, $p(x \mid i=1)$ and $p(x \mid i=2)$. The Bayes Rule says:

$$P(Decision=1 \mid X=x)=\frac{p(x \mid i=1)P(i=1)}{p(x \mid i=1)P(i=1)+p(x \mid i=2)P(i=2)} \tag{5.2}$$

and

$$P(Decision=2 \mid X=x)=\frac{p(x \mid i=2)P(i=2)}{p(x \mid i=1)P(i=1)+p(x \mid i=2)P(i=2)} \tag{5.3}$$

The Bayes Rule shows how the value of x changes the *a priori* probability $P(i=1)$, $P(i=2)$ to the *a posterior* probability $P(Decision=1 \mid x)$ and $P(Decision=2 \mid x)$.

If we have an observation x for which $P(Decision=1 \mid x)$ is greater than $P(Decision=2 \mid x)$, we attribute x to class $/m/$. Similarly, if $P(Decision=2 \mid x)$ is greater than $P(Decision=1 \mid x)$, we attribute x to class $/n/$.

According to the Maximum A Posterior (MAP) rule, we need to decide

$$\hat{Decision}(x)=\arg\max_{i=1,2} P(Decision=i \mid X=x)=\arg\max_{i=1,2} p(x \mid i)P(i) \tag{5.4}$$

Assume $p(x \mid i)$ is multivariate normally distributed. Then

$$p(x \mid i) = \frac{1}{(2\pi)^{\frac{d}{2}} |C_i|^{\frac{1}{2}}} \exp(-\frac{1}{2}(x - \mu_i)^T C_i^{-1}(x - \mu_i)), i = 1,2 \qquad (5.5)$$

where x is a d-component column vector, μ_i is the d-component mean vector, and C_i is the d-by-d covariance matrix. We often abbreviate the above formula as: $p(x \mid i) \sim N(\mu_i, C_i)$.

It can be shown that the distribution of any linear combination of normally distributed random variables is again normal. In particular, if A is a d-by-n matrix and $y = A^T x$ is a n-component vector, then $p(y) \sim N(A^T \mu, A^T C A)$. In the special case where A is a unit-vector a_1, $y = a_1^T x$ is a scalar that represents the projection of x onto a line in the direction of a_1 and $a_1^T C a_1$ is the variance of x onto a_1. In general, knowledge of the covariance matrix allows us to calculate the dispersion of the data in any direction.

From the above, we know the optimal classification is

$$\hat{Decision}(x) = \arg\max_{i=1,2} P(Decision = i \mid X = x) = \arg\max_{i=1,2} p(x \mid i)P(i) = \arg\max_{i=1,2} \ln(p(x \mid i)P(i))$$

$$= \arg\max_{i=1,2} [-\ln((2\pi)^{\frac{d}{2}} |C_i|^{\frac{1}{2}}) - \frac{1}{2}(x - \mu_i)^T C_i^{-1}(x - \mu_i) + \ln(P(i))]$$

$$= \arg\max_{i=1,2} [-\frac{1}{2}\ln(|C_i|) - \frac{1}{2}(x - \mu_i)^T C_i^{-1}(x - \mu_i) + \ln(P(i))]$$

Then we define the discriminant function as follows:

$$\delta_i(x) = -\frac{1}{2}\ln(|C_i|) - \frac{1}{2}(x - \mu_i)^T C_i^{-1}(x - \mu_i) + \ln(P(i)), i = 1,2 \qquad (5.6)$$

When we discuss the application of linear discriminant analysis (LDA) and quadratic discriminant analysis (QDA) in section 5.4, we will return to this formula again.

5.2 Adding More Independent Components Into The Feature Parameter Set

When the feature parameters are statistically independent, there are some theoretical results that suggest the possibility of excellent performance.

Consider one two-class multivariate normal case where $p(x \mid i) \sim N(\mu_i, C)$, $i = 1,2$ Here we use a pooled-covariance matrix, which is defined in (3.3), to be the common covariance matrix. If the *a priori* probabilities are equal, it can be shown that the Bayes error rate is given by

$$P(error) = \frac{1}{\sqrt{2\pi}} \int_{r/2}^{\infty} \exp(-\frac{1}{2}u^2)du, \tag{5.7}$$

where r^2 is the squared Mahanobis distance

$$r^2 = (\mu_1 - \mu_2)^T C^{-1} (\mu_1 - \mu_2) \tag{5.8}$$

Thus, the probability of error decreases as r increases, approaching zero as r approaches infinity. In the independent case, $C = diag(\sigma_1^2,...,\sigma_d^2)$, and

$$r^2 = \sum_{j=1}^{d} (\frac{\mu_{j1} - \mu_{j2}}{\sigma_j})^2 \tag{5.9}$$

This shows how each feature contributes to reducing the probability of error. The most useful features are those for which the difference between the means is large relative to the standard deviations. However, no feature is useless if its means for the two classes differ. An

obvious way to reduce the error rate further is to introduce new, independent features. If r can be increased without limit, then the probability of error can be made arbitrarily small.

In general, if the performance obtained with a given set of features is inadequate, it is natural to add some new features, particularly the ones that can help separate the class pairs most frequently confused. Notice that more features will increase the cost and complexity of the classifier.

Even if the probabilistic structure were unknown, the Bayes risk could not possibly be increased by adding new features, and if the new features provide any additional information, the performance must improve [40].

The New Features

The Combined Spectra Method in Chapter 4 uses the spectral information derived from both the nasals and the adjacent vowels as feature representations to set up a PCA model so that the PCA scores can be used to identify the nasals. We can see that all the 44 variables in the feature vectors were derived from the spectrum. Meanwhile, the model does not make use of any known properties of /*m*/ and /*n*/.

The locations of formants and anti-formants between /*m*/ and /*n*/ are different; thus, we can consider them as new features. The relevant information has been listed in Chapter 1, which is summarized in the following table:

Table 5.1
Locations for anti-formant/formant of nasals

	/*m*/	/*n*/
Anti-formant	500-1000Hz (800Hz in center)	1500-2000Hz (1700Hz in center)
2nd formant	1000Hz	2000Hz
3rd formant	2000Hz	2700Hz

We notice the following:

1. Around 800Hz, there is an anti-formant for /*m*/, but nothing for /*n*/
2. Around 1000Hz, there is an (2nd) formant for /*m*/, but nothing for /*n*/
3. Around 2000Hz, there is a (3rd) formant for /*m*/ and an (2nd) formant for /*n*/. Notice there is also one anti-formant between 1500Hz to 2000Hz for /*n*/.
4. Around 2700Hz, there is nothing for /*m*/ but a (3rd) formant for /*n*/.

Hence, we introduce 4 more components to the feature vector, each representing the energy in the above 4 spectral regions of the single glottal pulse in the nasal-boundary window which is farther from the release point.

We also notice that the RMS (root mean square) energy is used successfully to classify /*s*/ and /*z*/ and other sounds [25], so we consider RMS as another new variable of the model.

The mean and variance values derived from SVD algorithm II in Chapter 3 are added as two more variables of the feature vector.

The last variable added is the total energy in the range of 4000Hz and 11025Hz of the nasal-boundary window that is farther from the release point. This high frequency information is useful when we identify the voiced fricatives [90].

Covariance and Correlation

After adding all these variables into the feature vector, we need to test the independence of all the variables. Here we will use correlations instead of the covariance matrix for the following reasons:

The covariance of two features measures their tendency to vary together, but the coefficient of covariance is sensitive to the units of measurement; the coefficient of correlation remains invariant with respect to change of measurement unit.

The primary meaning of the coefficient of correlation lies in the amount of variation in one variable that is accounted for by the variable it is correlated with, so the coefficient shows the dependence between the two variables. Correlations between .00 and .30 are generally considered negligible; those between .30 and .70 are moderate; coefficients between .70 and 1.00 are considered high.

For more about the difference between covariance and correlation, see [91].

5.3 Dimensionality Reduction and Number of Training Samples

Dimensionality Reduction

In practical multi-category applications, it is very usual to encounter problems involving many features. In the Combined Spectra Method, there are 44 features; in the scenarios discussed in section 5.2, there are more than 44 features. One usually believes that each feature is useful for at least some of the discriminations. However, the features may not be independent, so some of them may be superfluous.

Consider a d-by-d correlation matrix $R = \{\rho_{ij}\}$, where the correlation coefficient ρ_{ij} is related to the covariance by

$$\rho_{ij} = \frac{\sigma_{ij}}{\sqrt{\sigma_{ii}\sigma_{jj}}} \qquad (5.10)$$

Since $0 \le \rho_{ij}^2 \le 1$, with $\rho_{ij}^2 = 0$ for uncorrelated features, and $\rho_{ij}^2 = 1$ for completely correlated features, ρ_{ij}^2 shows the role of a similarity function for features. Two features with large ρ_{ij}^2 are good candidates to be merged into one feature, thereby reducing the dimensionality by one.

Applying this idea to the features results in the following hierarchical procedure:

1. Let the initial dimension sizes be $\hat{d} = d + 1$, $d' = d$ and the set of features be $F_i, i = 1,...,d$; scale the features so that their numerical ranges are comparable.

2. If $\hat{d} = d'$, stop.

3. Let $\hat{d} = d'$. Compute the correlation matrix and find the pair of distinct features, say F_i, F_j that have a correlation larger than 0.7.

4. Merge F_i, F_j by taking the average between them, and delete F_j; decrease the number of dimensions by one, and let $d' = \hat{d} - 1$

5. Go to 2, repeat the process until all the correlation values are less than 0.7.

Merging using averaging assumes that the features have been scaled so that their numerical ranges are comparable.

One important aspect of variations must be mentioned at this time. The greatest emphasis is usually placed on those features or groups of features that have the greatest variability. However, in classification, we are interested in "discrimination," not representation. In other words, the most interesting features are the ones for which the difference in the class means is large relative to the standard deviations, not the ones for which only the standard deviations are large. This can be seen from (5.11). In this sense, discriminant analysis and logistic regression are better classification techniques than PCA theoretically.

Number of Training Samples

In practice, it can often be observed that adding new dimensions leads to worse rather than better performance (See Figure 3.5 and 3.6). This conflicts with the theoretical

conclusions in section 5.2. The basic source of this problem is the fact that the number of samples is finite, so "Curse of Dimensionality" occurs (Chapter 9 of [25]). We must have adequate samples in the training phase so that when we add new features, "Curse of Dimensionality" does not happen. However, for the purpose of software design, we have to use as few training samples as possible. Obviously, we need to establish a compromise on the number of training samples.

In this section, we look for a way to use a small number of samples to build an effective classifier.

Consider the covariance matrix, which can be estimated by the maximum likelihood estimate

$$\hat{C} = \frac{1}{n}\sum_{j=1}^{n} (x_j - \mu)(x_j - \mu)^t \qquad (5.11)$$

(For small n, say $n \leq 20$, we should replace the factor $(1/n)$ by $(1/(n-1))$ in order to produce an unbiased estimator.)

It is the sum of n independent d by d matrices of rank one, and thus is singular if $n < d$ [40]. Since we need the inverse of the covariance matrix to obtain the discriminant functions, the number of samples must be at least $d+1$.

To smooth out statistical fluctuations and obtain a good estimate, it is reasonable to have several times of that number of samples when building the classifier. Unfortunately, we are required not to have many samples in the training phase.

In our case, because there are a large number of dimensions in the original feature vector, the number of available training samples is often inadequate since it may be no larger than the number of dimensions, which is at least 44. Fortunately, one can reduce the dimensionality, either by refining the features, by selecting a good subset of the existing features (such as by merging), or by combining the existing features in some way (such as

PCA or factor analysis). Another possibility is to assume that the two classes, /m/ and /n/, share the same covariance matrix, and to pool all the available data (We often do that in PCA and LDA) [40]. In our experiments, merging, as we described previously, and pooling are used.

About Training and Testing Data Set

In our experiments, training and testing samples are generated at one time and they are put together into a data set. Then, the question is: given a fixed number of samples, how many samples should be used in training phase and how many in the testing phase? Here we are concerned about the number of samples needed in the training phase. Consequently, there would be new questions that need further discussions. If most of the data is used in training, then one cannot have confidence in the test; if most of the data is used in testing, then one cannot obtain a good classifier. There is no definitive answer to this partition problem.

Fortunately, there are ways to solve this problem. In this thesis, the following procedure is designed when only a limited number of samples are available.

Procedure for Limited Number of Training Samples

1. Calculate the feature vector, $\hat{x}$, which include components calculated from combined spectra, RMS value, energy around formants or anti-formants, energy in high frequencies, mean and variance values in SVD algorithm II. The dimension of $\hat{x}$ is 52. The features may not be independent.
2. Merge the features according to the algorithm introduced below (5.10) so that there are a smaller number of dimensions in the feature vector. Let this number be d and new feature vector be $\tilde{x}$. We know $d \leq 52$.
3. Examine the samples in the current data set:
   Let the number of testing samples be s (In my experiments, I let $s = 1$). Then the number of training samples is $n - s$. Select different combinations of samples for this $(n - s, s)$ partition. Implement statistical classification techniques using the resulted sample combinations, evaluate and then average the

performance. By this way, we can achieve the performance while using $n \square s$ training samples.

4. Repeat this process, each time decreasing the amount of training samples by s. Stop until the number of training samples is equal to $d+1$.
5. Plot a graph showing the relationship between the number of training samples and its corresponding averaged classification scores.

In Step 3, we use a special way to solve the data partition problem. We name the sounds so that they are such a sequence in the data folder (Assume we only deal with three vowels: /a/, /i/, /u/):

"am1, im1, um1, am2, im2, um2, am3, im3, um3, am4, im4, um4, …."

When the number of training sounds is 5, the different combinations are taken using the following way (Similar to the way we generate the rows for singular value matrix in Chapter 4):

{ am1, im1, um1, am2, im2}, { im1, um1, am2, im2, um2}, { um1, am2, im2, um2, am3}, …

In this way, we use approximately the same number of different vowel-nasal transition sounds in the training data so that each kind of vowel-nasal sound makes the same contribution to the establishment of the training model.

Using this procedure, we can determine how the performance changes when the number of training samples changes. From the graphs, we can also determine how many training samples we need for a desired classification score.

In the following sections, we introduce statistical classification methods dealing with experiments using fixed numbers of training and testing samples. The Procedure for Limited Number of Training Samples is implemented for each method (except PCA, explained in section 5.7) so that there is one graph on averaged classification scores for each method.

5.4 PCA with 52 Parameters

In section 3.2, we summarized the algorithm of PCA classification. In this section, we use the same algorithm, changing only the size of feature vectors. After we add the eight parameters described in section 5.2, the size becomes 44+8=52.

The experimental results are summarized in section 5.7.

5.5 Discriminant Analysis

In linear DA (LDA), we find linear combinations of the quantitative variables that provide maximal separation between the classes or groups. This method maximizes the ratio of between-class variance to the within-class variance in the data set so it can guarantee maximal separability [39]. The main difference between DA and PCA is that PCA does more feature classification, whereas DA does more data classification. In PCA, the shape and the location of the original data sets changes when transformed to a different space; DA does not change the location, but tries to provide more class separability and draw a decision region between the given classes [92][93].

Under different assumptions, several versions of DA algorithms can be used.

Classical LDA and QDA

When $p(x \mid i)$ is multivariate normally distributed, the classification scores for DA can be expressed using (5.6).

If the C_i, $i = 1,2$, are both equal to a covariance matrix C, we ignore the first term in (5.6). In this case, the classification boundaries are linear, so it is called Linear Discriminant Analysis. We usually use the pooled covariance matrix for C, which is defined as follows:

$$C = \frac{(N_1 - 1) \cdot C_1 + (N_2 - 1) \cdot C_2}{(N_1 - 1) + (N_2 - 1)}, \tag{5.12}$$

If the covariance matrices are not equal, there are no simplified scores and the method is called Quadratic Discriminant Analysis.

Results of LDA rely heavily on the assumption of equality of variance matrices. QDA fits the data better than LDA, but has more parameters to estimate. Notice if the data are not multivariate normal, LDA and QDA may miss useful classification patterns.

Class-dependent and Class-independent LDA

There are two other versions of LDA in the literature [92]. In thoseLDAs, training data sets are transformed and testing data sets are classified in the transformed space by two different approaches.

1. Class-dependent transformation: This type of approach involves maximizing the ratio of between-class variance to within-class variance. The main objective is to maximize this ratio so that adequate class separability is obtained. The class-specific type approach involves using two optimizing criteria for transforming the data sets independently.

2. Class-independent transformation: This approach involves maximizing the ratio of overall variance to within-class variance. This approach uses only one optimizing criteria to transform the data sets, and hence, all data points, irrespective of their class identity, are transformed using this transform. In this type of LDA, each class is considered a separate class from all other classes.

The mathematical operations are as follows:

1. Compute the mean of training data set of each class and the mean of the entire (pooled) training data set. Let μ_1 and μ_2 be the means of training sets of /*m*/ and /*n*/, respectively, and μ_3 be the mean of the pooled training data, which is obtained in the following way:

$$\mu_3 = P(i=1)\mu_1 + P(i=2)\mu_2 \tag{5.13}$$

2. For a single class problem, the within-class scatter is the class covariance, C_1 or C_2, while for a two–class problem, the within-class scatter is obtained as follows:

$$S_w = P(i=1)\cdot C_1 + P(i=2)\cdot C_2 \tag{5.14}$$

Between-class scatter is computed using the following equation:

$$S_b = (\mu_1 - \mu_3)\cdot(\mu_1 - \mu_3)^T + (\mu_2 - \mu_3)\cdot(\mu_2 - \mu_3)^T \tag{5.15}$$

The optimizing criterion in LDA is the ratio of between-class scatter to the within-class scatter. The solution obtained by maximizing this criterion defines the axes of the transformed space.

3. If the LDA is a class-dependent type, for a two-class problem, one separate optimizing criterion is required for each class. The optimizing factors in the case of class dependent type are computed as

$$Criterion_i = C_i^{-1} * S_b \tag{5.16}$$

For the class independent transform, the only optimizing criterion is computed as

$$Criterion = S_w^{-1} * S_b \quad (5.17)$$

4. By definition, an eigenvector of a transformation represents a 1-D invariant space of the vector space in which the transformation is applied. A set of these eigenvectors whose corresponding eigenvalues are non-zero are all linearly independent and are invariant under the transformation. Thus, any vector space can be transformed in terms of linear combinations of the eigenvectors. A linear dependency between features is indicated by a zero eigenvalue. To obtain a non-redundant set of features, all eigenvectors corresponding to non-zero eigenvalues only are considered and the ones corresponding to zero eigenvalues are ignored [19]. In the case of LDA, the transformations are found as the eigenvector matrix of the different criteria defined in the above.

   For any two class problem we would always have one non-zero eigenvalue. This is attributed to the constraints on the mean vectors of the classes. The eigenvector corresponding to the non-zero eigenvalue works for the definition of the transformation.

   Having obtained the transformation matrices, we transform the training data using the single transform or the class specific transforms, whichever the case may be.

   For the class dependent LDA,

$$y_i = A_i \cdot x_i, i = 1,2 \quad (5.18)$$

   where A_i is the class specific transform matrix derived from $Criterion_i$ in (5.16), x_i is the training data for class i, and y_i is the LDA - transformed training data.

   For the class independent LDA,

$$y = A \cdot x \tag{5.19}$$

where A is the transform matrix derived from *Criterion* in (5.17), x is the pooled training data, and y is the LDA- transformed pooled training data.

5. Similarly, the testing data vectors are transformed and then classified using the Euclidean distances between the testing vectors and the class means.

   Euclidean distance is computed using equation (5.25). For a two class problem, two Euclidean distances are obtained for each testing point.

   For the class dependent LDA,

$$r_i = \left\| A_i \cdot x - u_i' \right\|, i = 1,2 \tag{5.20}$$

   where x is the testing data, A_i is the same matrix as in (5.23), u_i' is the mean of class i derived by averaging the transformed training data of class i, r_i is the distance from the transformed testing data to u_i'.

   For the class independent LDA,

$$r_i = \left\| A \cdot x - u_i'' \right\|, i = 1,2 \tag{5.21}$$

   where x is one testing data, A is the same matrix as in (5.24), u_i'' is the mean of class i derived by averaging the transformed training data of class i, r_i is the distance from the transformed testing data to u_i''.

   The smaller Euclidean distance between the two distances classifies the testing vector.

5.6 Support Vector Machine

The next classification technique we use is the Support Vector Machine (SVM). In this thesis, we introduce only the basic ideas of SVM and main results on adaptive tuning. For further information, refer to [45] and [49] by Dr. Zhang Hao.

SVM was first used for classification from the early 1990s. From then on, it soon became the method of choice for many researchers and practitioners involved in supervised machine learning. It was found that the SVM could be derived as the solution to an optimization problem in a Reproducing Kernel Hilbert Space (RKHS) [49], thus bearing a resemblance to penalized likelihood and other regularization methods used in nonparametric regression. This served to link the rapidly developing SVM literature in supervised machine learning to the now obviously related statistics literature. The question why SVM works well theoretically was answered in [49], where it was shown that, provided a rich enough RKHS is used, the SVM implements the Bayes Rule for classification. An examination of the form of the SVM shows that it is doing the implementation in a flexible and particularly efficient manner.

On the topic of pattern recognition, SVMs have been used for isolated handwritten digit recognition, object recognition, speaker identification, charmed quark detection, face detection in images, and text categorization (For references on those applications, please refer to the Introduction in [49]). In most of these cases, performance (i.e., error rates on test sets) of SVM either matches or is significantly better than that of competing methods.

The basic idea ofSVMs is that, roughly speaking, for a given learning task, with a given finite amount of training data, the best generalization performance will be

achieved if the right balance is struck between the accuracy attained on that particular training set, and the capacity of the machine, that is, the ability of the machine to learn any training set without any error. For more details on those concepts, please refer to the theory of statistical learning [94].

As with other regularization methods, there are always one or even several tuning parameters that must be chosen well in order to have efficient classification in nontrivial cases. In this thesis, we use one method called the Generalized Approximate Cross Validation ($GACV$) for parameter tuning [45][46][47][48].

In the following, we talk about SVMs in the two-category problem. We implement the standard case, where the training set is representative of the general population and the cost of misclassification is the same for both categories. We do not consider the nonstandard case, where neither of these assumptions is valid.

SVMs

For SVMs, the data, x, is coded as follows:

$$\begin{aligned} g = +1 &\text{ if } x \text{ is attributed to class } /m/ \\ g = -1 &\text{ if } x \text{ is attributed to class } /n/ \end{aligned} \qquad (5.22)$$

The support vector optimization problem is:

$$\begin{aligned} &\text{Find } f(x) = b + h(x) \text{ with } h \in H_k \text{ to} \\ &\text{minimize } \frac{1}{n}\sum_{i=1}^{n} (1 - g_i f(x_i))_+ + \lambda \|h\|^2_{H_k} \end{aligned} \qquad (5.23)$$

where n is the number of samples; x_i is the component of x; $(\tau)_+ = \tau$, if $\tau > 0$, and 0 otherwise; $\lambda > 0$; b is one vector; H_k is the reproducing kernel Hilbert space (RKHS) with reproducing kernel

$$K(p,q),\ p,q \in \Omega \qquad (5.24)$$

(For more on RKHS, see [98]). Assume the minimizer of (5.23) is f_λ. The classifier is computed as follows:

$$\begin{aligned} &\text{If } f_\lambda(x) > 0 \text{, then } g = +1 \\ &\text{If } f_\lambda(x) < 0 \text{, then } g = -1 \end{aligned} \tag{5.25}$$

For this SVM classifier, we were motivated to say that it is optimally tuned if λ minimizes a proxy for the Generalized Comparative Kullback-Liebler distance ($GCKL$), which is defined as

$$GCKL(\lambda) = E_{true} \frac{1}{n} \sum_{i=1}^{n} (1 - g_{new,i} f_\lambda(x_i))_+ \tag{5.26}$$

That is, λ and possibly other parameters in K are chosen to minimize a proxy for an upper bound on the misclassification rate.

The $GACV$ for choosing λ and other parameters in K

The goal here is to obtain a proxy for the $GCKL(\lambda)$ in (5.26). See details at [45].

Let $f_\lambda^{[-k]}$ be the minimizer of the form $f = b + h$ with $h \in H_k$ to

$$\text{minimize } \frac{1}{n} \sum_{\substack{i=1 \\ i \neq k}}^{n} (1 - g_i f(x_i))_+ + \lambda \|h\|_{H_k}^2 \tag{5.27}$$

Let

$$V_0(\lambda) = \frac{1}{n} \sum_{i=1}^{n} (1 - g_k f_\lambda^{[-k]}(x_k))_+ \tag{5.28}$$

We can write

$$V_0(\lambda) \equiv OBS(\lambda) + D(\lambda), \tag{5.29}$$

where

$$OBS(\lambda) = \frac{1}{n}\sum_{i=1}^{n}(1 - g_k f_\lambda(x_k))_+ \qquad (5.30)$$

and

$$D(\lambda) = \frac{1}{n}\sum_{i=1}^{n}[(1 - g_k f_\lambda^{[-k]}(x_k))_+ - (1 - g_k f_\lambda(x_k))_+] \qquad (5.31)$$

It can be showed that $D(\lambda) \approx \hat{D}(\lambda)$, where

$$\hat{D}(\lambda) = \frac{1}{n}[2\sum_{g_i f_\lambda(x_i)<1}\frac{1}{2n\lambda}K(x_i,x_i)\delta_i + \sum_{-1\le g_i f_\lambda(x_i)\le 1}\frac{1}{2n\lambda}K(x_i,x_i)\delta_i] \qquad (5.32)$$

where $K_{n\times n} = \{K(x_i,x_j)\}$. In this study, we are using a Gaussian kernel, which has the expression $K(p,t) = \exp(-\frac{\|p-t\|^2}{2\sigma^2})$. The parameters λ and σ are jointly tuned by the criterion $GACV$.

The $GACV$ is defined as

$$GACV(\lambda) = OBS(\lambda) + \hat{D}(\lambda) \qquad (5.33)$$

5.7 Experimental Results

We conducted experiments for voice samples of two speakers, who are named “DB” and “DW” in Chapter 3.

Each voice sample contains five glottal pulses segmented from the transition regions of nasal – vowel boundary. For each sample, we generate different kinds of features as previously described so that the dimension of the feature vector is 52.

For each sample we have in hand (including both training data and testing data), we standardize each component of the corresponding feature vector and then test for

independence. We hope to find the same pattern of dependence between (groups of) features for different speakers. However, the results are discouraging.

For example, in syllable-initial case, for speaker DW's samples, the following group of features show the most independence:

"Combined spectra values in Bark 2, Bark 16, Bark 33; RMS value;
mean and variance values"

For speaker DB's samples, the group of features are:

"Combined spectra values in Bark 3, Bark 4, Bark 16, Bark 17, Bark 26, Bark 36, Bark 38;
energy at 2700Hz; mean and variance values"

We can see that only three features, Bark 16 energy, mean and variance values, show the same strong independence for both speakers. So we abandon any attempt to find a uniform independent feature set for different speakers.

Now let us consider the following two tables.

Table 5.2
Data for Syllable-initial Case for Two Speakers Used in DA and SVM

Category	# of data of /*m*/	# of data for /*n*/	# of independent features	Range of # of training data for /*m*/	Range of # of training data for /*n*/
Speaker DB	109	88	42	46-80	46-80
Speaker DW	87	82	39	46-80	46-80

Table 5.3
Data for Syllable-final Case for Two Speakers Used in DA and SVM

Category	# of data for / *m* /	# of data for / *n* /	# of independent features	Range of # of training data for / *m* /	Range of # of training data for / *n* /
Speaker DB	61	190	40	43-55	43-55
Speaker DW	64	186	35	43-55	43-55

In the above tables, Columns II and III show the number of available data for each nasal sound, respectively. Column IV shows the number of independent features after testing of independence. According to (5.11), the possible numbers of training data for each class, which are shown in Columns V and VI, must be larger than the numbers in Column IV.

At this time we need to point out that since the voice samples contain differentvowels, the training and testing data are composed of different nasal-vowel windows. Thus, we are dealing with a pool of data containing different kinds of vowels.

PCA

PCA in Section 5.4 is the only method we do not implement "Procedure for Limited Number of Training Samples" on. We experiment using this method on a fixed number of training data. Because of the properties of PCA, it is difficult to decide the number of eigenvalues that we should keep in order to obtain a best classification score. This uncertainty makes PCA an impractical technique, and we only examine it as a method in the sense of research. PCA can not be implemented into the automated system.

Table 5.4
Data for Syllable-initial Case for Two Speakers Used in PCA Method

NASALS	/*m*/		/*n*/	
Category	Number of Training data of /*m*/	Number of Testing data for /*m*/	Number of Training data for /*n*/	Number of Testing data for /*n*/
Speaker DB	87	22	70	18
Speaker DW	44	43	47	35

Table 5.5
Data for Syllable-final Case for Two Speakers Used in PCA Method

NASALS	/*m*/		/*n*/	
Category	Number of Training data for /*m*/	Number of Testing data for /*m*/	Number of Training data for /*n*/	Number of Testing data for /*n*/
Speaker DB	51	10	100	90
Speaker DW	45	19	91	95

Table 5.6
Performance of PCA in Syllable-initial Case

	Closed Test for /*m*/	Closed Test for /*n*/	Open Test for /*m*/	Open Test for /*n*/
DB	100%	95%	92%	84%
DW	100%	100%	66%	83%

Table 5.7
Performance of PCA in Syllable-final Case

	Closed Test for /*m*/	Closed Test for /*n*/	Open Test for /*m*/	Open Test for /*n*/
DB	100%	100%	87%	82%
DW	84%	100%	75%	77%

In the closed tests, the classification scores are approximately 100% in most cases, which means the training model is well established. In the open tests, the classification score is acceptable for speaker DB, but not high for speaker DW. Notice those scores are the best ones we can obtain by keeping different number of eigenvalues.

DA

Classical LDA and QDA, and class independent LDA do not work well in the experiments. They do not show a less than 20% misclassification rate even when the maximal possible number of training data are used.

Class - dependent LDA and SVM work better. See the following figures:

Class – dependent LDA

Figure 5.1
Performance of LDA in Syllable-initial Case on Speaker DB
The maximal accuracy is 84%, obtained when using 80 training samples

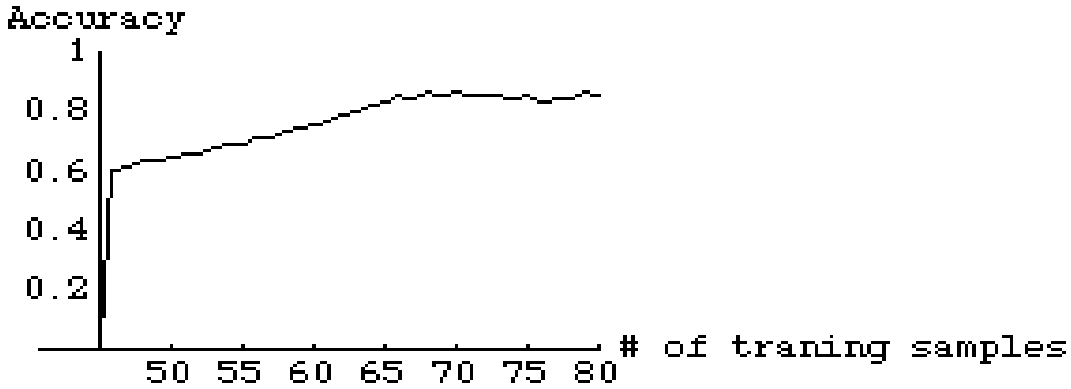

Figure 5.2
Performance of LDA in Syllable-initial Case on Speaker DW
The maximal accuracy is 86%, obtained when using 67 training samples

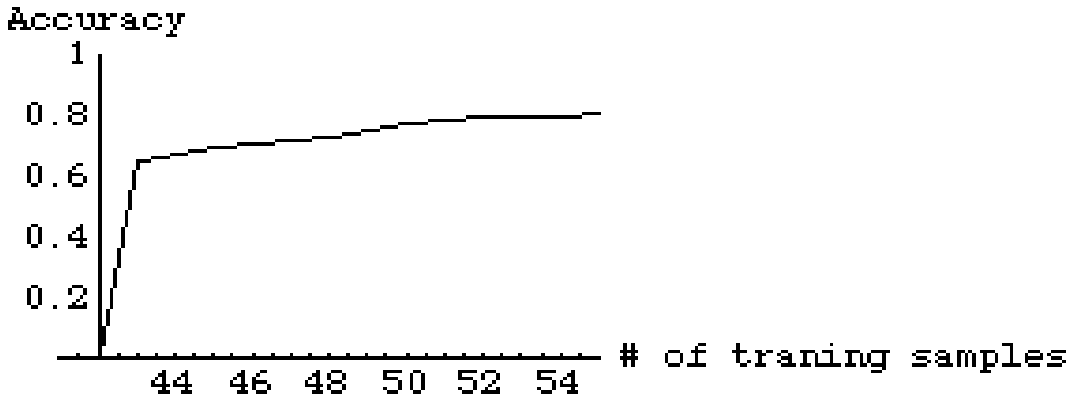

Figure 5.3
Performance of LDA in Syllable-final Case on Speaker DB
The maximal accuracy is 77%, obtained when using 54 training samples

Figure 5.4
Performance of LDA in Syllable-final Case on Speaker DW
The maximal accuracy is 82%, obtained when using 52 training samples

From those figures, we can see that we need at least 65 training samples for syllable initial – case, and at least 55 training samples for syllable – final case for a less than 20% misclassification rate.

SVM

SVM with *GACV* tuning works well too. The misclassification rate is less than 18.5% if at least 60 training samples for both syllable–initial and syllable–final cases are used.

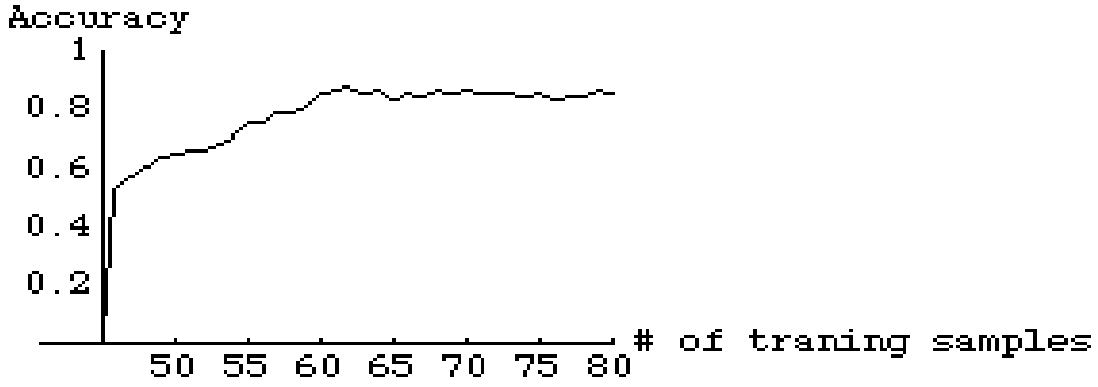

Figure 5.5
Performance of SVM in Syllable-initial Case on Speaker DB
The maximal accuracy is 84%, obtained when using 62 training samples

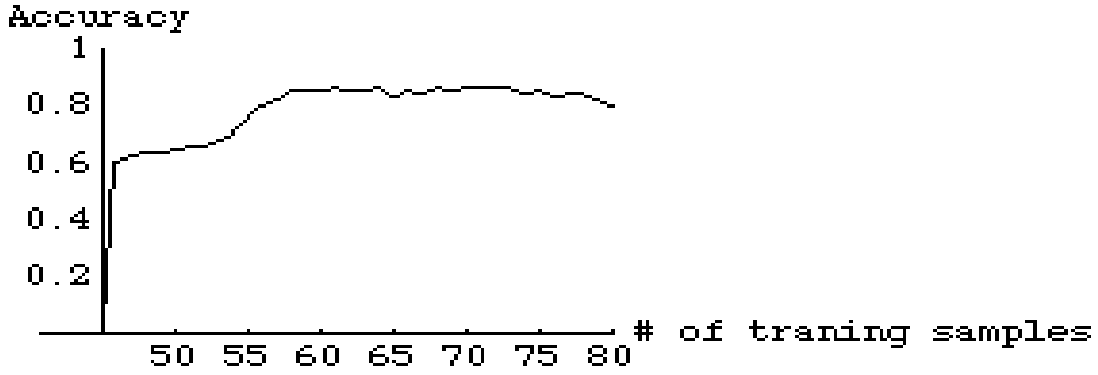

Figure 5.6
Performance of SVM in Syllable-initial Case on Speaker DW
The maximal accuracy is 82%, obtained when using 60 training samples

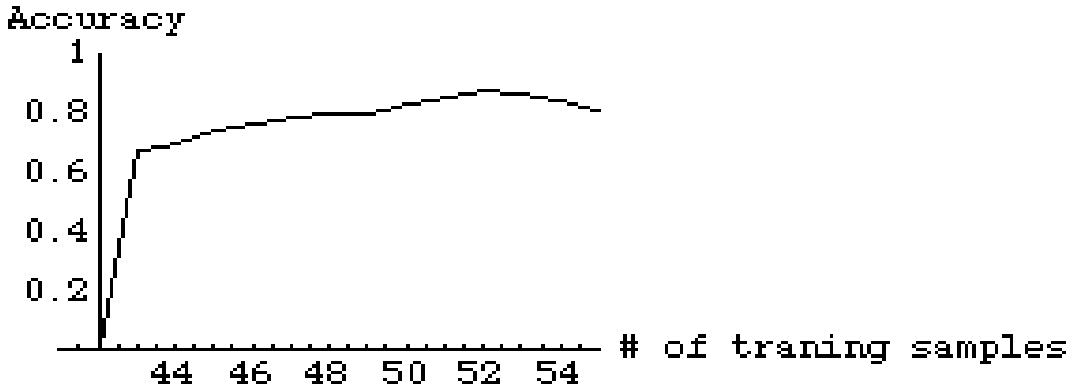

Figure 5.7
Performance of SVM in Syllable-final Case on Speaker DB
The maximal accuracy is 82%, obtained when using 53 training samples

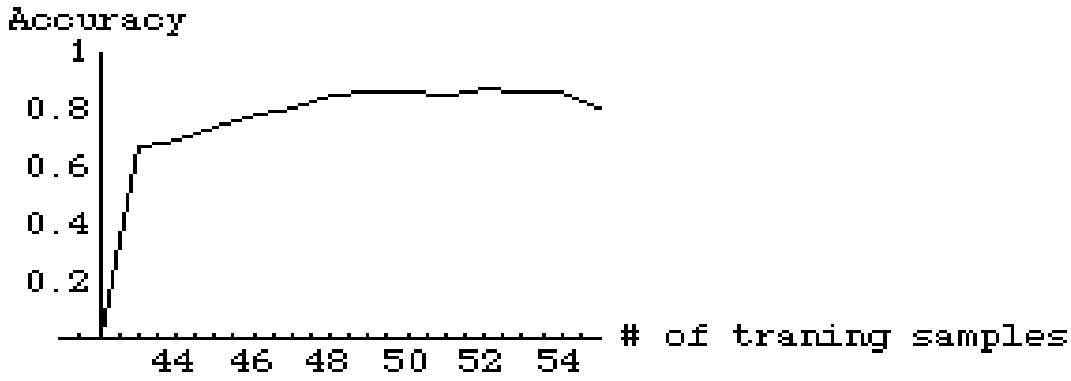

Figure 5.8
Performance of SVM in Syllable-final Case on Speaker DW
The maximal accuracy is 81%, obtained when using 52 training samples

Chapter 6

Conclusions and Future Study

In this thesis, we summarized the recent work on the topic of identifying betweennasal consonants. We have made valuable trials on feature extraction and we have conducted experiments using the most popular statistical classification methods, such as DA and SVM. Finally, we concluded that we need about 50-60 training samples to establish a reliable classifier with low (Less than 20%) misclassification rate. We have also tried to perform the analysis automatically using the Glottal Pulse Tracker, instead of segmenting samples manually each time. In fact, we observe that the GP tracker produces spikes wherever the release point occurs; again, using the GP tracker, we automatically locate the five glottal pulses we need for the algorithms. So the process can be done automatically. But for some unknown reasons, all the classification methods result in nearly 50% misclassification rate, even when we used the maximal possible number of training samples. It implies that the GP tracker produce different results from those produced manually. This automation problem needs further study.

In the feature extraction step, there are some other DSP transforms that can be used to generate useful information. The wavelet transform is one of the most popular transforms dealing with non-stationary signals. In [80], wavelets were used for feature extraction on phoneme recognition and it can achieve 100% accuracy with a 60% confidence level. One important step of this method is to choose the most suitable wavelet dictionaries for the problem. It is usually done by choosing the wavelet that gives the minimal entropy among the available wavelet dictionaries [95]. However, in our problem, the waveforms of both nasals are similar in the time domain, which makes it very difficult to select proper wavelets to represent them. The applications of the wavelet transform on nasals will be an interesting topic of future study.

There are still many reliable statistical classification methods in the literature. For example, logistic regression [59][60][61][62] does not need the assumption of the Normal distribution; K-Nearest Neighbor (KNN) classifier and neural network are useful methods too. After more statistical analysis and learning further about the distribution of nasal data, it is reasonable to determine a best classification method to handle this problem. This is for future study, also.

References:

1. Jonathan Harrington. The contribution of the murmur and vowel to the place of articulation distribution in nasal consonants. *Journal of Acoustic Society of America*, 96 (1), July, 1994.

2. Kurowski, K., and Blumstein, S.E. Perceptual integration of murmur and formant transitions for place of articulation in nasal consonants. *Journal of Acoustic Society of America*, 76, 383-390, 1984.

3. Kurowski, K., and Blumstein, S.E. Acoustic properties for place of articulation in nasal consonants. *Journal of Acoustic Society of America*, 81, 1917-1927, 1987.

4. Fromkin, V., Rodman, R., and Hyams, N. *An Introduction to Language*, Seventh Edition. Boston, MA:Heinle, A Division of Thomson Learning Inc., pp ix-620, 2003.

5. Rodman, R., McAllister, D., Bitzer, D., Cepeda, L., and Abbitt, P. Forensic Speaker Identification Based on Spectral Moments. *Forensic Linguistics*, 9(1), pp 22-43, July, 2002.

6. Rodman, R. Linguistics and the Law: How Knowledge of, or Ignorance of, Elementary Linguistics May Affect the Dispensing of Justice. *Forensic Linguistics*, 9(1), pp 92-101, July, 2002.

7. Wang, M., Bitzer, D., McAllister, D., Rodman, R., Taylor, J. An Algorithm for V/UV/S Segmentation of Speech. *Proceedings of the 2001 International Conference on Speech Processing (ICSP'2001). Seoul, Korea: Acoustic Society of Korea (ASK)*, pp 541-546. September, 2001.

8. Taylor, J., Bitzer, D., Rodman, R., McAllister, D., Wang, M. Speaker Independence in Lip Synchronization of Vowels and Distinguishing between /m/ and /n/. *Proceedings of the 5th World Multiconference on Systemics, Cybernetics and Informatics (SCI 2001) and the 7th International Conference on Information Systems Analysis and Synthesis (ISAS 2001). Skokie, IL:International Institute of Informatics and Systemics, August, 2001*.

9. Krothapalli, C., McAllister, D., Rodman, R. Bitzer, D., Wang, M., and Taylor, J. Predictor Surfaces for Lip Synchronization Animation Of Voiced Input. *Proceedings of the 5th World Multiconference on Systemics, Cybernetics and Informatics (SCI 2001) and the 7th International Conference on Information Systems Analysis and Synthesis (ISAS 2001). Skokie, IL:International Institute of Informatics and Systemics, August, 2001*. BEST PAPER IN SESSION AWARD.

10. Rodman, R. Computer Recognition of Speakers Who Disguise Their Voice. *Proceedings of the International Conference on Signal Processing Applications & Technology 2000 (ICSPAT2000),* October, 2000.

11. Rodman, R., McAllister, D., Bitzer, D., and Chappell, D. A High-Resolution Glottal Pulse Tracker. *Proceedings of* the *International Conference on Spoken Language Processing*. *(ICSLP2000), October, 2000.*

12. Rodman, R., McAllister, D., Bitzer, D., Fu, H. and Xu, B. A Pitch Tracker for Identifying Voiced Consonants. *Proceedings of the 10th International Conference on Signal Processing Applications and Technology* (ICSPAT'99). November, 1999.

13. Fu, H., Rodman, R., McAllister, D., Bitzer, D. and Xu, B. Classification of Voiceless Fricatives through Spectral Moments. *Proceedings of the 5th International Conference on Information Systems Analysis and Synthesis* (ISAS'99). Skokie, IL:International Institute of Informatics and Systemics, pp 307-311, 1999.

14. McAllister, D., Rodman, R., Bitzer, D. and Freeman, A. Speaker Independence in Automated Lip-Sync for Audio-Video Communication. *Computer Networks & ISDN Systems,* V 30, No 21-22, pp 1975-1980, 1998.

15. Rodman, R., McAllister, D., Bitzer, D. and Freeman, A. Automated Lip-Sync Animation as a Telecommunications Aid for the Hearing Impaired. *Proceedings of the Conference on Interactive Voice Technology for Telecommunications Applications* (IVTTA'98). Published by the European Speech Community Association (ESCA) and the IEEE, pp 204-209, 1998.

16. Rodman, R. Speaker Recognition of Disguised Voices. *Proceedings of the Consortium on Speech Technology Conference on Speaker Recognition by Man and Machine: Directions for Forensic Applications,* M. Demirekler, A. Saranli, H. Altincay, and A. Paoloni (eds). Ankara, Turkey:COST250 Publishing Arm, pp 9-22, April, 1998.

17. Rodman, R., McAllister, D. and Bitzer, D. Lip Synchronization of Speech. *Proceedings of the Audio-Visual Speech Processing Conference '97* (AVSP'97), pp 133-136, 1997.

18. Rodman, R., McAllister, D. and Bitzer, D. Lip Synchronization as an Aid to the Hearing Disabled. *Proceedings of the American Voice Input/Output Society,* pp 233-248, 1997.

19. Golub & Van Loan. *Matrix Computations*, 3rd edition.

20. John R. Cameron, James G. Skofronick, Roderick M. Grant. *Physics of The Body*, 1992.

21. Richard E.Berg, David G.Stork. *The Physics of Sound*, 1982.

22. Fant, G. *The Acoustic Theory of Speech Production*, The Hague: Mouton, 1960.

23. Flanagan, J.L. *Speech Synthesis, Analysis, and Perception*, New York: Springer-Verlag, 1972.

24. Charles E. Speaks. *Introduction to Sound: Acoustics for the Hearing and Speech Sciences*, 3rd, 1999.

25. Jonathan Harrington, Steve Cassidy. *Techniques in Speech Acoustics*, 1999.

26. Philip F. Seitz, Marianne M. McCormick, Ian M. C. Watson, R. Anthony Bladon. Relational spectral features for place of articulation in nasal consonants. *Journal of the Acoustical Society of America*, VOL. 87, NO. 1, January, 1990, pp 351-358.

27. Lester C.M. Chan, Y.S. Cheung. Analysis and Recognition of Isolated Putonghua Vowels by Karhunen-Loeve Transformation Techniques. *Speech Communications*, Vol.5, 1986, page 299-330.

28. Lawrence R.Rabiner, Ronald W.Schafer. *Digital Processing of Speech Signals*, 1978.

29. John R.Deller, John H.L.Hansen, John G.Proakis. *Discrete-Time Processing of Speech Signals*, 2000.

30. John G.Proakis. *Digital Signal Processing: Principles, Algorithms, and Applications*, 3rd, 1996.

31. C. Radhakrishna Rao. *Linear Statistical Inference and Its Applications*, 2nd, 1973.

32. Dendrinos et al., *Speech Commun*. 10, 45--57 (1991).

33. Lawrence K.Saul, Jont B.Allen. Periodic Component Analysis: An Eigenvalue Method for Representing Periodic Structure in Speech. *Advances in Neural Information Processing Systems,* 13, 2001.

34. Seitz, P.F, McCormick, M.M., Watson, I.M.C, and Bladon, R.A. Relational spectral features for place of articulation in nasal consonants. *J.Acoust.Soc.Am*.87, 351-358, 1990.

35. Jeih-weih, Hung, Hsin-min Wang, Lin-shan Lee. Automatic Metric-based Speech Segmentation for Broadcast News via Principle Component Analysis, 2000 (*Obtained from the web*).

36. Tim Kientzle. A Programmer's Guide to Sound, 1997.

37. Alan Agresti. *An Introduction to Categorical Data Analysis*, 1996.

38. Alan Agresti. *Categorical Data Analysis*, 1990.

39. Anant M.Kshirsagar. *Multivariate Analysis*, 1983.

40. Richard O.Duda, Peter E.Hart. *Pattern classification and Scene Analysis*, 1973.

41. Lawrence Rabiner, Biing-Hwang Juang. *Fundamentals Of Speech Recognition*, 1993.

42. Thomas W.Parsons. *Voice And Speech Processing*, 1986.

43. John R.Deller,Jr., John H.L.Hansen, John G.Proakis. *Discrete-time Processing of Speech Signals*, 2000.

44. L.R.Rabiner, R.W.Schafer. *Digital Processing of Speech Signals*, 1978.

45. Wahba, G., Lin, Y., Lee, Y. and Zhang, H. Optimal Properties and Adaptive Tuning of Standard and Nonstandard Support Vector Machines. *TR 1045*, Oct 2001. *Prepared for the Proceedings of the MSRI Berkeley Workshop on Nonlinear Estimation and Classification*. This TR supercedes TR 1039.

46. Wahba,G.,Lin,Y,.Lee,Y.,Zhang,H. On the Relation between GACV and Joachims' \xi\alpha Method for Tuning Support Vector Machines, With Extension to the Nonstandard Case. *TR 1039*, June 2001.

47. Lin,Y.,Wahba,G.,Zhang,H. and Lee,Y. Statistical Properties and Adaptive Tuning of Support Vector Machine. *TR 1022*, Sep 2000. Has appeared in *Machine Learning*, 48, 115-136, 2002.

48. Wahba, G.,Lin,Y. and Zhang,H. Generalized Approximate Cross Validation for Support Vector Machines, or, Another Way to Look at Margin-Like Quantities *TR 1006*, April 1999. `*Advances in Large Margin Classifiers*, Smola, Bartlett, Scholkopf and Schurmans,eds. *MIT Press* (2000), 297-309, 2000.

49. Christopher J.C.Burges. A Tutorial on Support Vector Machines for Pattern Recognition. *Data Mining and Knowledge Discovery 2*: 121-167, 1998.

50. Nello Cristianini. Support Vector and Kernel Machines. *http://www.support-vector.net/tutorial.html*.

51. Peilv Ding, Liming Zhang. Speaker Recognition Using Principal Component Analysis (Obtained from the web).

52. Harvey E.Rhody. Digital Image Processing and Pattern Recognition. Notes for SIMG-784, Winter Quarter, 1997.

53. I.Potamitis, N.Fakotakis, G.Kokkinakis. Independent Component Analysis Applied to Feature Extraction for Robust Automatic Speech Recognition (Obtained from the web).

54. K.Ducinskas, J.Saltyte. Quadratic Discriminant Analysis of Spatially Correlated Data. Nonlinear Analysis: Modeling and Control, v.6, No.2, 15-28, 2001.

55. Gupta P.L, Riley J.T., White T.J. Misclassification Probabilities for Quadratic Discriminant. SIAM.Sci.Stat.Comput., 7(4), 1986.

56. Jain A.K, Duin R.P.W., Mao J. Statistical Pattern Recognition: A Review. IEEE Transactions on Pattern Analysis and Machine Intelligence, 22(1), 4-38, 2000.

57. Wakaki H. Comparison of Linear and Quadratic Discriminant Functions. Biometrica, 77, 227-229, 1990.

58. Lawoko C.R.O., McLachlan G.J. Discrimination with Auto-correlated Observations. Pattern Recognition, 18(2), 145-149, 1985.

59. Ying So. A Tutorial on Logistic Regression (Obtained form the web).

60. Hosmer, D.W., Lameshow, S. Applied Logistic Regression. Wiley, New York, 1989.

61. Strauss, D. The Many Faces of Logistic Regression. The American Statistician, vol 46, No.4, pp. 321-326, 1992.

62. Paul Komarek, Andrew Moore. Logistic Regression for Data Mining, Text Classification, Link Detection, and Large Datasets (Obtained from the web).

63. Classification Methods. Klumer Academic Publishers.

64. Matthew A. Siegler, Uday Jain, Bhiksha Raj, Richard M.Stern. Automatic Segmentation, Classification and Clustering of Broadcast News Audio (Obtained from the web).

65. Adriano Petry, Adriano Zanuz, Dante Augusto Couto Barone. Bhattacharyya Distance Applied to Speaker Identification (Obtained from the web).

66. Ronald R.Coifman, Mladen Victor Wickerhauser. Entropy-based Algorithms for Best Basis Selection. IEEE Transactions on Information Theory, vol.38, No.2, March, 1992.

67. Ronald R.Coifman, Mladen Victor Wickerhauser. Best-adapted Wavelet Packet Bases. Yale Univ., Feb 1990. (Obtained from the web).

68. I.Daubechies. Ortho-normal Bases of Compactly Supported Wavelets. Communications of Pure & Applied Mathematics, vol. XLI, pp. 909-996, 1988.

69. J.S.Milton, Jesse C.Arnold. Introduction to Probability and Statistics: Principles and Applications for Engineering and the Computing Sciences, 3rd, 1995.

70. Qi Li, Frank K.Soong, Olivier Siohan. A High Performance Auditory Feature for Robust Speech Recognition, 2000. (Obtained from the web).

71. Julius O.Smith. Bark and ERB Bilinear Transforms. IEEE Transactions Speech and Audio Processing, vol. 7, No. 6, November, 1999.

72. Ren-cang Li. Relative Perturbation Theory: Eigenvalue and Singular Value Variations. SIAM J. Matrix Ana. Appl, vol. 19, No. 4, pp. 956-982, October 1998.

73. Jacob Benesty. Adaptive Eigenvalue Decomposition Algorithm for Passive Acoustic Source Localization. Acoustical Society of America, 2000.

74. Fuad Gwadry, Richard Harshman, Mary Varga, Allen Braun. Singular Value Decomposition of PET Images.

75. Simon Doclo, Ioannis Dologlou, Marc Moonen. A Novel Iterative Signal Enhancement Algorithm for Noise Reduction in Speech. Proceedings of the 5th International Conference on Spoken Language Processing (ICSLP'98), Sydney, Australia, pp 1435-1438, Dec, 1998.

76. Russell H.lambert, Marcel Joho, Heinz Mathis. Polynomial Singular Values for Number of Wideband Sources Estimation and Principal Component Analysis (Obtained from the web).

77. Michael E.Wall, Andreas Rechtsteiner, Luis M.Rocha. Singular Value Decomposition and Principal Component Analysis. A Practical Approach to Micro-array Data Analysis, 2002.

78. Ross Cutler. Face Recognition Using Infrared Images and Eigenfaces, April, 1996 (Obtained from the web).

79. James S.Walker. A Primer on Wavelets and Their Scientific Applications, 2000.

80. C.J. Long, S.Datta. Wavelet Based Feature Extraction for Phoneme Recognition, 1998 (Obtained from the web).

81. Kadambe S, Srinivasan. Applications of Adaptive Wavelets for Speech. Optical Engineering, 33(7), pp. 2204-2211, July, 1994.

82. Kadambe S, Boudreaux-Bartels. Applications of the Wavelet Transform for Pitch Detection of Speech Signals. IEEE Transactions on Information Theory, vol. 38, pp. 917-924, March, 1992.

83. Szu H, Telfer B, Kadambe S. Neural Network Adaptive Wavelets for Signal Representation and Classification. Optical Engineering, vol. 31, No.9, pp. 1907-1916, September, 1992.

84. Saito N. Local Feature Extraction and Its Application Using a Library of Bases. Phd thesis, Yale University, 1994.

85. Zeev Litichever, Dan Chazan. Classification of Transition Sounds with Application to Automatic Speech Recognition. Euro-speech, Scandinavia, 2001.

86. R. R. Coifman, F. B. Geshwind, and F. Warner. Discriminant feature extraction using empirical probability density estimation and a local basis library. *Pattern Recognition*, vol. 35, pp.2841-2852, 2002.

87. Classification of geophysical acoustic waveforms and extraction of geological information using time-frequency atoms. *1996 Proc. Computing Section of Amer. Statist. Assoc.*, pp.322-327, 1997.

88. R.R.Coifman. On local feature extraction for signal classification. *Applied Analysis* (O. Mahrenholtz and R. Mennicken, eds.), special issue of *Zeitschrift fur Angewandte Mathematik und Mechanik*, pp.453-456, Akademie-Verlag, Berlin, 1996.

89. R.R.Coifman. Selection of best bases for classification and regression. *Proc.1994 IEEE-IMS Workshop on Information Theory and Statistics*, p.51, IEEE-IMS, Oct.1994, Alexandria, VA.

90. Henry Fu. *Classification of Voiceless Fricatives Through Spectral Moments*, thesis, 1999.

91. Covariance and Correlation. *http://www.public.asu.edu/~kkelley/edp502/edp6.htm*

92. S.Balakrishnama, A.Ganapathiraju. Linear Discriminant Analysis – A Brief Tutorial (*Obtained from the web*).

93. J.Duchene, S.Leclercq. An optimal Transformation for Discriminant and Principal Component Analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 10, No. 6, November, 1988.

94. V.Vapnik. *Statistical Learning Theory*. John Wiley and Sons, Inc., New York, 1998.

95. Buckheit, Donoho. Wavelet and Reproducible Research. *Wavelets and Statistics*, Springer – Verlag, New York, 1995.

96. Lip Syncing Project. *http://www.multimedia.ncsu.edu/research/voiceio/*

97. The Bark Scale. *http://www-users.york.ac.uk/~pgc104/phonlink/bark.html*

98. G.Wahba. Spline Models for Observational Data. *SIAM*, 1990. *CBMS-NSF Regional Conference Series in Applied Mathematics*, v. 59,

Appendices

A Description of the Code and Data Samples Used in the Thesis

The data sets, Mathematica and Matlab codes used in Chapter 4 and 5 are saved in a CD-ROM named "Wang Thesis".

In this CD-ROM, there are two folders: "PDF files" and "Programs and speech samples". "PDF files " contains a copy of the thesis with pdf format. "Programs and speech samples" contains two folders: "Chapter 4 experiments" and "Chapter 5 experiments". In each of those folders, programs and data sets for each algorithm in Chapter 4 and 5 are saved. Each algorithm has its own folder, in which the corresponding data sets, Mathematica or Matlab codes are saved. The names of those algorithm folders are as follows:

"SVD1", "SVD2", "EVD4", "LDA", "SVM".

Most programs are written in Mathematica. Only the SVM algorithm is written in Matlab. In the beginning of each file, there is a header to explain what it is. In addition, the important steps are well-described using comment line.

When a reader would like to access certain kind of technique or graph described in the thesis, as long as he has read the thesis for at least once and knows which chapter the technique is in, by following the naming scheme in the CD-ROM, he can reach the related Mathematica notebooks or Matlab .m files and data sets very easily.

For example, in order to access the notebook file that generates Figure 4.11a, we should go to folder "Chapter 4 experiments", then "SVD2", look for the file named "SVD2_rodman_ahm.nb". Run the file and Figure 4.11a will be generated.