uOttawa

L'Université canadienne
Canada's university

FACULTÉ DES ÉTUDES SUPÉRIEURES ET POSTOCTORALES

uOttawa

L'Université canadienne
Canada's university

FACULTY OF GRADUATE AND POSDOCTORAL STUDIES

Cheng Wu

AUTEUR DE LA THÈSE / AUTHOR OF THESIS

M.Sc. (Systems Science)

GRADE / DEGREE

Systems Science

FACULTÉ, ÉCOLE, DÉPARTEMENT / FACULTY, SCHOOL, DEPARTMENT

A Typology for Voice and Music Signals

TITRE DE LA THÈSE / TITLE OF THESIS

Mayer Alvo

DIRECTEUR (DIRECTRICE) DE LA THÈSE / THESIS SUPERVISOR

CO-DIRECTEUR (CO-DIRECTRICE) DE LA THÈSE / THESIS CO-SUPERVISOR

EXAMINATEURS (EXAMINATRICES) DE LA THÈSE / THESIS EXAMINERS

Andre Dabrowski

Mahmoud Zarepour

Gary W. Slater

LE DOYEN DE LA FACULTÉ DES ÉTUDES SUPÉRIEURES ET POSTDOCTORALES /
DEAN OF THE FACULTY OF GRADUATE AND POSTDOCORAL STUDIES

University of Ottawa

Master's Program in Systems Science

Thesis

A Typology for Voice and Music Signals

By
Cheng Wu B.Sc

A Thesis
Submitted to the School of Graduate Studies and Research
In partial fulfillment of the requirements
for the degree of
Master of Science in Systems Science

Your file Votre référence
ISBN: 0-494-11457-6
Our file Notre référence
ISBN: 0-494-11457-6

Abstract

With the high increase in the availability of digital music, it has become of interest to automatically query a database of musical pieces. At the same time, a feasible solution of this objective gives us an insight into how humans perceive and classify music.

In this research, we discuss our approach to classify music into four categories: pop, classical, country and jazz. Songs are collected in wave format. We randomly chose five 10-second clips from different parts of a song. We discussed two families of features: wavelet features and time-based features. These features are capable of capturing the information of energy and time of voice signal. Instead of using traditional Mel-Frequency Cepstral Coefficients (MFCC)[7] methods, which are widely used in audio classification and music classification, we incorporate the features in statistical classification methods such as LDA, QDA and tree. Finally, we attempted to create an adaptive tree approach for classification.

In this research, 130 songs are collected. Pop songs are collected in 4 languages, English, Chinese, Spanish and French. A cross validation method is used to compute the proportion of correctly classified songs. It is shown that the tree method has a proportion of correct classification equal 0.80 when pop and country are considered as one category.

Acknowledgements

I would like to thank my supervisors Professor Mayer Alvo and Professor François Théberge for all their help and advice. Also I would like to thank Professor Dabrowski and Professor Zarepour for their help and comments.

Table of Content

List of Tables

List of Figures

Chapter 1 Introduction

Problem

It is common in psychology and in the social sciences to classify data into types. For example, psychologists may be interested in looking at personality types whereas archeologists may be concerned with the classification of old English bottles by time period. The data for these problems are usually finite dimensional vectors and for such data, standard multivariate statistical methods are available. For example, in the case of old English bottles, one may look at various size measurements whereas in the case of personality types we would be considering categorical variables. It is of interest to ask whether a typology can be determined for voice and music signals. Such a typology if it exists, could potentially result in quantifying more precisely:

1) The category to which a given song belongs;
2) The changes over time occurring in the song of a specific artist.

Applications abound. For example:

- The evolution and voice maturity of an artist could be better documented.
- A song may be more objectively placed into existing categories
- The variety in a singer's repertoire can be assessed and compared to other artists.

- The classification process can be automated; it could be used to document the extensive musical libraries that exist today.

The principal goal of this thesis is to develop a mathematical algorithm for classifying a musical voice signal. Complex data such as voice and music signals are difficult to deal with. A typical song lasting 5 minutes would require about 500 megabytes to store in the wave format. Consequently it is more efficient to store characteristic features of a song. Although Fourier analysis can be used to decompose a signal, it does not take into account the local behavior or features of a song. This is particularly important since most songs contain frequent high and low intonations. Consequently it is necessary to look for other methods.

One method that holds promise is the use of wavelets to decompose the signal in multiple time and frequency (or details) components. One objective in this thesis is to make use of features that arise from wavelet decomposition and to include them in a general statistical analysis of voice and music.

There are two main contributions of this thesis:

1. To compile a music database consisting of song clips for various artists in each of four music categories, such as classic, jazz, country and pop;
2. To research the usefulness of two types of features for the purpose of classification. The first type of features originates from the use of wavelet decomposition whereas the second is based on the time series of the songs. We make use of both these features for the classification into categories.

Chapter 2 Background

In this chapter, we are going to introduce some concepts of voice and file format that we use in the project. It will help us to understand this project better.

2.1 Basic concepts

What is sound?

A vibration of air brought about by pressure creates sound. The vibration presses the air molecules around it and this is transmitted through the air medium. Our ears hear the vibrations as sound. (see for example [1])

What is a wave?

A sound wave describes the variation in air pressure. The sound we hear is usually made up of several sound waves mixed together. Those waves may be in phase (same frequency and peaks), 180 degrees out of phase (canceling each other) or somewhere in between.

How is sound recorded?

A microphone is often used to record sound to analog audio, such as tape. A microphone converts air pressure to voltage, high pressure represents high voltage, and low pressure represents low voltage. Speakers do the opposite and convert voltage to air pressure.

A computer system can store digital data and a sound card converts voltage to digital audio. A sound card samples voltage at usually twice the rate of the highest frequency.

Example 1: The frequency of a voice is 8000 HZ and it needs 16000 samples per second; This is called the Nyquist sampling frequency.

Example 2: CD quality sound is 44100 samples per second for digital audio.

2.2 The wave file format

Music is frequently stored on a computer as a wave. In this format file, it is not compressed. Wave files utilize a PCM (pulse code modulation) format that converts analog into digital signals. An analog signal is one whose amplitude and/or frequency vary continuously with time. Information in analog form cannot be processed by a digital computer and consequently, it is necessary to convert it into digital form. Digital data can be transported robustly over long distances unlike the analog data and can be interleaved with other digital data so various combinations of transmission channels can be used. In what follows, the term "digital" will apply to the encoding technique, which means digitalization of analog information in general. Hence a PCM encoded signal is nothing more than streams of bits.

In order to illustrate how voltage data is coded in PCM format, suppose that the voltage ranges from –2 to 2 volts. Suppose that we wish to use 4 bits to code where a bit is either a "0" or a "1". Four bits can represent any number from 0 to 15. Hence, a voltage interval is calculated to D = 2*2/2^4 = 0.275 volts. As a result, 1 in PCM format represents 0.275 volts. The table in Appendix 4 shows the transformation of voltage to PCM code.

In Figure 2-1 below, we exhibit a sample of a wave file from the song "A new day has come". In Appendix 2, details are provided for how to interpret this information. Hence, we can deduce the following information:

Total File Size: 45,508,652 bytes

Format: PCM (pulse code modulation)

Data sample frequency: 44,100 Hz,

Bits Per Sample: 16

Channels: 2

Data Size: 45,508,608 bytes

Play Time: 257.985 seconds

Total Samples: 11,377,152

Average Bytes Per Second: 176,400

Block Alignment: 4

```
000000  52 49 46 46 24 68 B6 02  57 41 56 45 66 6D 74 20  RIFF$h..WAVEfmt
000010  10 00 00 00 01 00 02 00  44 AC 00 00 10 B1 02 00  ........D.......
000020  04 00 10 00 64 61 74 61  00 68 B6 02 00 00 00 00  ....data.h......
000030  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
000040  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
000050  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
000060  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
000070  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
000080  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
000090  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
```

Figure 2-1 A sample of wave file

2.3 Programs used for the project

A number of different software programs were used in the thesis. Cool Editor was used to convert from a CD format to a wave format. Visual C++ was used for data sampling whereas R and MINITAB were used for the statistical analysis. Below we describe each of these.

Cool Editor

We purchased the Cool Editor package from the Syntrillinm Company. Cool Editor is an audio software used to record music, voice, or other audio as well as to edit and mix it with other audio or musical pieces. It can read and write in different formats, such as mp3, pcm (wave) or to convert files from one format to another. In Figure 2-2, we have reproduced a clip from applying Cool Edit 2000 to a musical piece from the singer Holy Cole. The waveform for each of the two channels can be seen and there are options to cut, copy and paste.

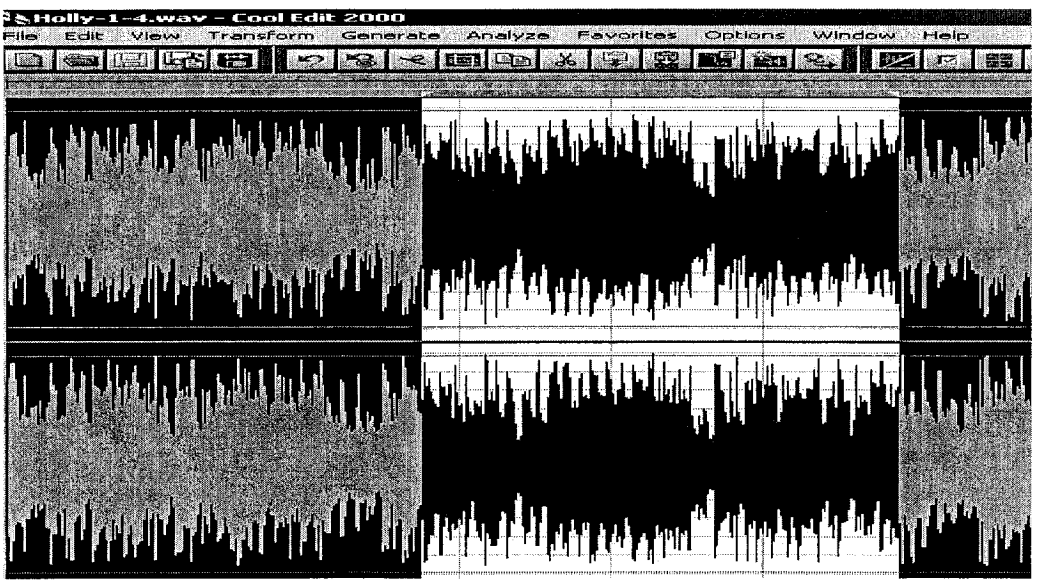


Figure 2-2 Snapshot of the use of Cool Editor

Visual C++

Visual C++ is a high level language for developing application programs. A high level language permits the user to write understandable code by exploiting the language structure to reach the objective. Visual C++ can be used for data processing, producing low-level drivers for hardware, databases and animation applications, network communications, file transformations and user interfaces. In this thesis, C++ has been used principally for data processing. Figure 2-3 exhibits an example of the interface of Visual C++ 6.0. The left window shows the files in the project whereas the right window points to the corresponding source code.

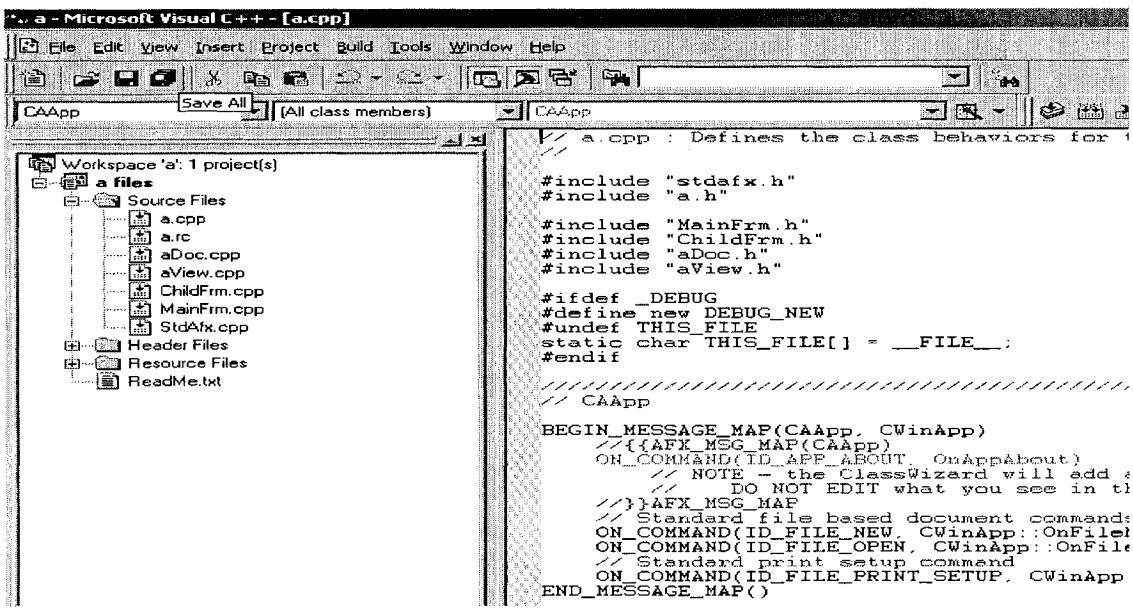


Figure 2-3 Snapshot of the use of Visual C++

R

R is an open source language similar to S, which was developed at Bell Laboratories (formerly AT&T, now Lucent Technologies) by John Chambers and his colleagues. It is used for statistical computing and graphics applications. There are some important differences with S, but much of the code written for S runs unaltered under R.

R provides a wide variety of statistical (linear and nonlinear modeling, classical statistical tests, time-series analysis, classification, clustering) and graphical techniques, and is highly extensible. The S language is often the vehicle of choice for research in statistical methodology, and R provides an open source route to participation in that activity [5][2]. Figure 2-4 below exhibits a snapshot from a session using R.

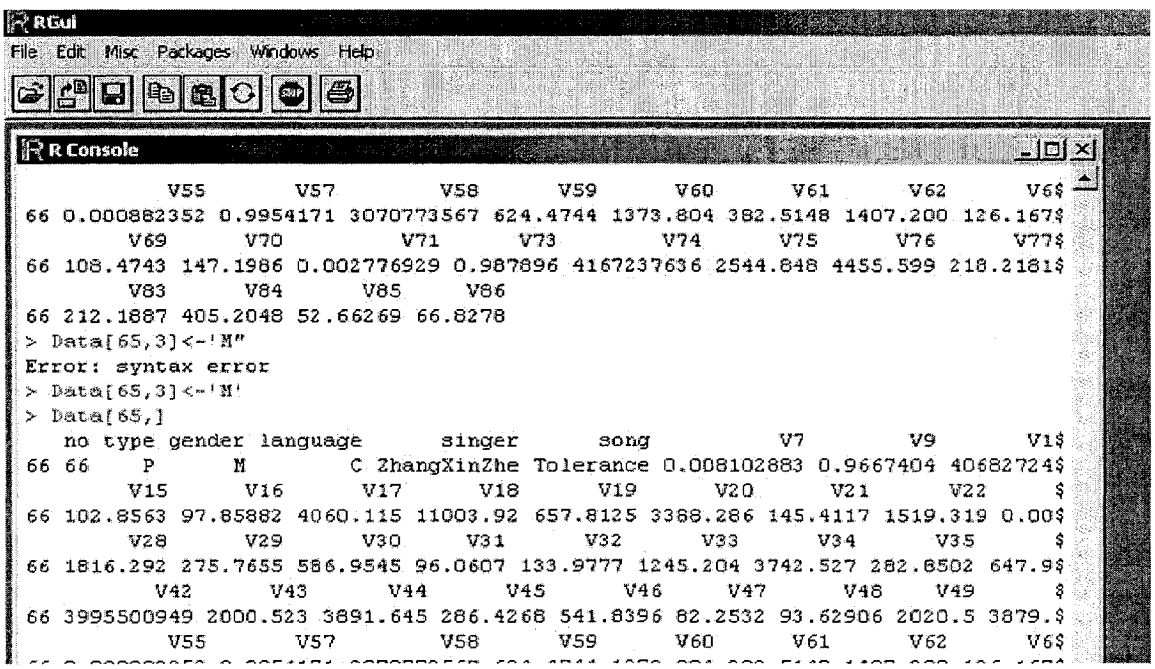
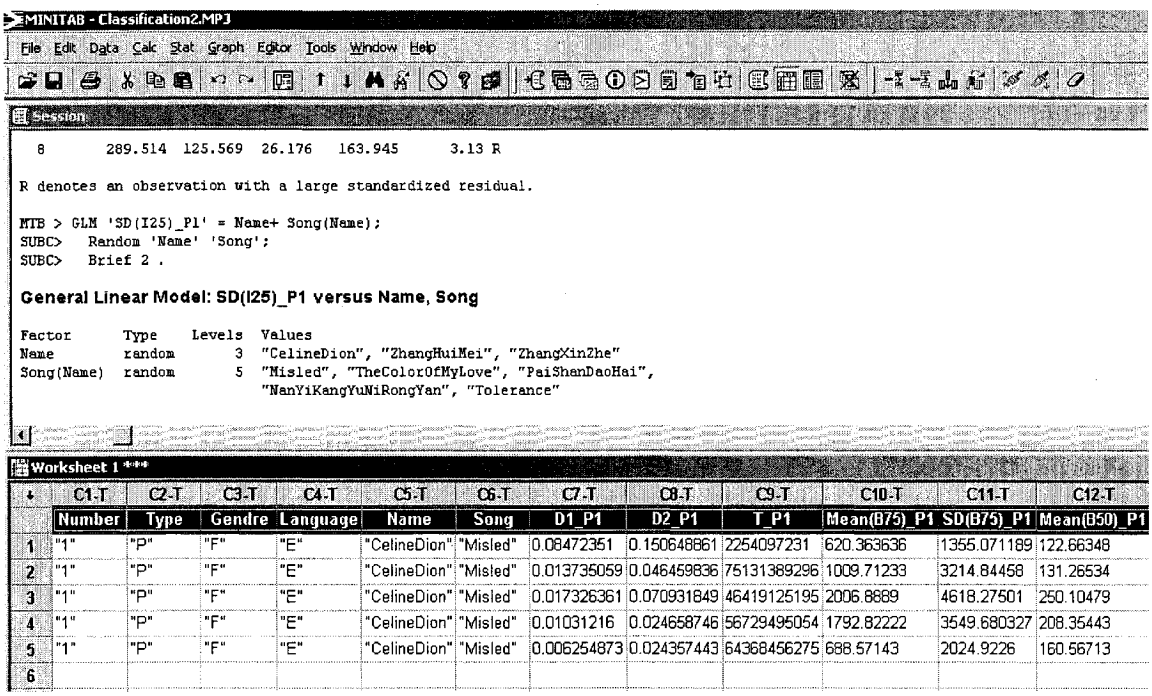


Figure 2-4 Snapshot from the use of R

MINITAB

MINITAB is a powerful, easy-to-use, statistical software package that provides a wide range of basic and advanced data analysis capabilities. Figure 2-5 exhibits a snapshot of a session using MINITAB.



MINITAB - Classification2.MPJ

File Edit Data Calc Stat Graph Editor Tools Window Help

Session

```
  8       289.514  125.569  26.176   163.945      3.13 R

R denotes an observation with a large standardized residual.

MTB > GLM 'SD(I25)_P1' = Name+ Song(Name);
SUBC>   Random 'Name' 'Song';
SUBC>   Brief 2 .

General Linear Model: SD(I25)_P1 versus Name, Song

Factor       Type    Levels  Values
Name         random       3  "CelineDion", "ZhangHuiMei", "ZhangXinZhe"
Song(Name)   random       5  "Misled", "TheColorOfMyLove", "PaiShanDaoHai",
                              "NanYiKangYuNiRongYan", "Tolerance"
```

Worksheet 1 ***

	C1-T	C2-T	C3-T	C4-T	C5-T	C6-T	C7-T	C8-T	C9-T	C10-T	C11-T	C12-T
	Number	Type	Gendre	Language	Name	Song	D1_P1	D2_P1	T_P1	Mean(B75)_P1	SD(B75)_P1	Mean(B50)_P1
1	"1"	"P"	"F"	"E"	"CelineDion"	"Misled"	0.08472351	0.150648861	2254097231	620.363636	1355.071189	122.66348
2	"1"	"P"	"F"	"E"	"CelineDion"	"Misled"	0.013735059	0.046459836	75131389296	1009.71233	3214.84458	131.26534
3	"1"	"P"	"F"	"E"	"CelineDion"	"Misled"	0.017326361	0.070931849	46419125195	2006.8889	4618.27501	250.10479
4	"1"	"P"	"F"	"E"	"CelineDion"	"Misled"	0.01031216	0.024658746	56729495054	1792.82222	3549.680327	208.35443
5	"1"	"P"	"F"	"E"	"CelineDion"	"Misled"	0.006254873	0.024357443	64368456275	688.57143	2024.9226	160.56713
6												

Figure 2-5 Snapshot from the use of MINITAB

Chapter 3 Database

In this thesis, we collected songs in four principal categories: Pop, Jazz, Country and Classic. The categories used corresponded to those in use by the music section in the bookstore Chapters. Popular singers were arbitrarily chosen in each category and songs were selected from several of their CDs. For the category Pop, songs were selected in English, French, Spanish and Mandarin Chinese. The complete database is given in Appendix 1.

3.1 Building the database

The process used to build the database consists of 3 steps, as shown schematically in Figure 3-1. We describe each step next.

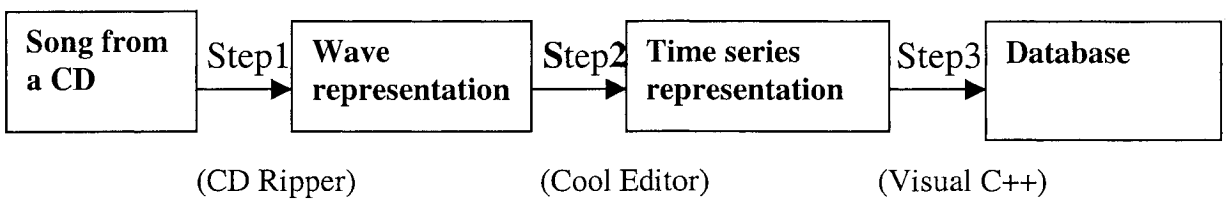


Figure 3-1 Process of building the database

Step 1: From CD format into wave file format

We collect songs of different types from CDs. Then we download the free software CD Ripper from [6]. We used the ripper software to convert from the CD format to wave files, which can then be stored on a computer. We collected 130 songs from 34 singers. (See Appendix 1)

Step 2: From wave file to time series

A wave file for a single song can easily occupy over 500 megabytes. We used Cool Editor to convert each wave file into a time series and obtained a vector of numbers in plain ASCII format, which can then be read by statistical packages such as MINITAB or R.

Step 3: Sampling from time series

We convert the wave files to a time series, which are then saved as text files. Since such a file can occupy 100M to 200M on disk, we decided to randomly select five 10-second clips from each song for further analysis.

3.2 Sampling from time series

Figure 3-2 represents a schematic diagram to indicate the algorithm that was used for sampling of the musical clips.

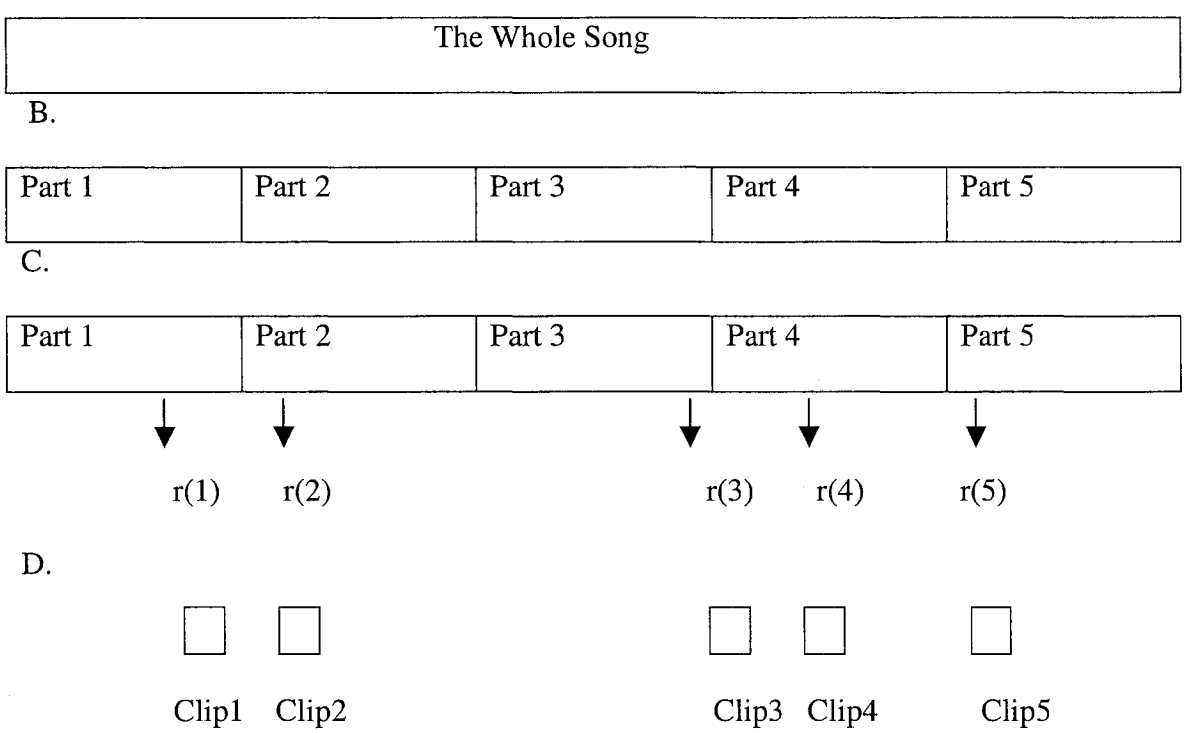


Figure 3-2 Algorithm of sampling from time series

We use Visual C++ 6.0 to write a program to sample data, which goes as follows:

A. Read a file and compute n(1), the sample size;

B. Divide the song into 5 parts, and let n(2)=n(1)/5;

C. From each part, generate a random number r(i) between (i-1)*n(2) to (i-1)*n(2)+t, where t is the sample size of a 10-second clip.

D. Produce five 10-second clips with respective starting points r(1), r(2), r(3), r(4), r(5).

Chapter 4 Wavelet Theory and Features

In this chapter, we apply a wavelet analysis to voice signals and define several wavelet-based features for classification. We also utilize some statistical methods to investigate the ability of the wavelet features to distinguish categories. The general methodology is described in section 4.1.

4.1 Methodology

Various songs were catalogued and converted to a digital signal as described in chapter 3. The general methodology proposed consists of decomposing the signal using a wavelet decomposition and then extracting various features from it. Those features will then be evaluated for their ability to discriminate among musical categories. The diagram in Figure 4-1 shows the methodology.

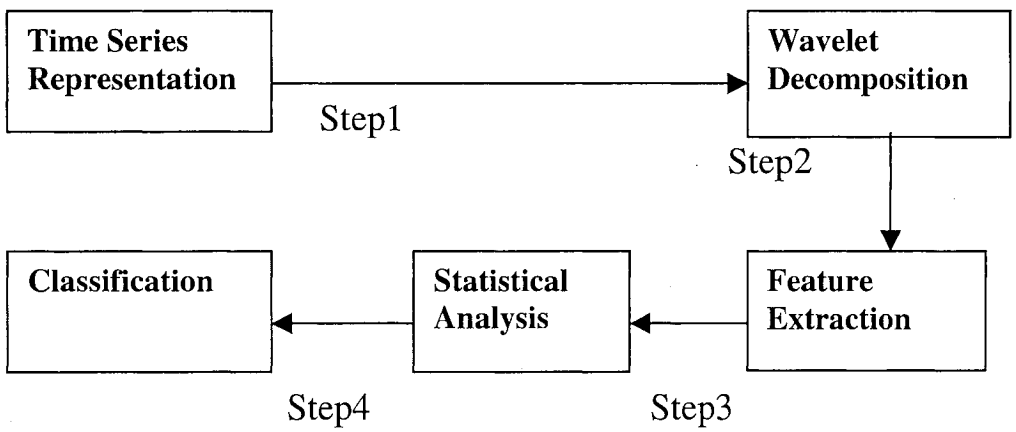


Figure 4-1 Methodology

Step 1: Wavelet Decomposition of the Signals

Wavelet decomposition divides the signal into two types of components: a smooth representation and several levels of details. We illustrate the decomposition of a simple signal (Doppler wave with Gaussian noise) later in Figure 4-2.

Step 2: Feature Extraction

The feature extraction step consists of identifying numerical characteristics that are potentially useful for statistical classification and clustering.

Step 3: Statistical Analysis

In order to test the usefulness of any given set of features for discriminating among categories, we can use analysis of variance (ANOVA) techniques. Specifically, we may for each feature, postulate a nested model where songs are nested within categories and test for differences among the categories. This global technique can be used to uncover quickly whether or not there are any differences among categories.

Step 4: Classification

The features selected in Step 3 can then be used in a linear discriminant analysis LDA, a quadratic discriminant analysis QDA or a tree-based method to classify songs into categories. Clustering techniques, such as hierarchical clustering or k-means can also be used.

4.2 Wavelets

After introducing the general methodology, we are going to explain wavelet theory briefly in this section. Wavelet transforms are useful tools for analyzing signals and images. Unlike Fast Fourier Transforms (FFTs), which provide analysis only in the frequency domain, wavelet decompositions also provide chronological (spatial for images) information. A detailed introduction on wavelets along with additional references may be found in [3].

Typical usages of wavelet decompositions include:

- Image or signal de-noising
- Image signal compression
- Feature gathering

In our context, the data consists of 1-dimensional discrete signals $f(t)$, and we will use a wavelet decomposition to gather features on those signals.

There are father wavelets and mother wavelets, which satisfy respectively:

$$\int_{-\infty}^{+\infty} \phi(t)dt = 1, \int_{-\infty}^{+\infty} \psi(t)\ dt = 0$$

The father wavelets represent the smooth, low frequency parts of a signal whereas the mother wavelets represent the detail high frequency parts. These base wavelets generate a whole family of such functions as follows:

$$\psi_{a,b}(t) = |a|^{-1/2}\ \psi((t-b)/a)$$

where a is the scale parameter and b the shift. The parameters a and b differ depending on the time scale that we are looking at, and the level of details (frequency).

Discrete wavelet transforms (DWTs) are typically explained either as a sequence of filters, or via the multi-resolution analysis (MRA). In the filters approach, the idea is to run the signal through a high-pass filter (which retains the high half of the frequencies) and a low-

pass filter. Since the signal coming out of the low-pass filter has at maximum half the frequency of the original signal, thus half of the samples can be discarded, and the same process can be repeated. At each step, the low-pass filter yields the smooth component of the signal, and the high-pass filter yields the high component of the signal and the details. In the MRA representation, we decompose the signal as follows.

Let f_k be a function in $V_k \subset L^2(R)$. We split f_k into its low and high frequency components (filters):

- smooth (low frequency) components f_{k+1} are orthogonal projections of f_k onto $V_{k+1} \subset V_k$.
- f_{k+1} gives a smoothed approximation of f_k. The lost information (the details) g_{k+1} is the projection of f_k onto W_{k+1}, where $V_k = V_{k+1} \oplus W_{k+1}$. W_{k+1} is known as the wavelet space.

From the above, we have that $f_k = f_{k+1} + g_{k+1}$. We can continue to decompose f_{k+1} in the same way. Given a sequence of subspaces and modulo a few technical conditions, all spaces have an orthogonal basis given by the system of the equations:

$$\phi_{j,k}(t) = 2^{-j/2}\phi(2^{-j}t - k) \qquad k, j \in Z$$

$$\psi_{j,k}(t) = 2^{-j/2}\psi(2^{-j}t - k) \qquad k, j \in Z$$

We can approximate $f(t)$ via:

$$f(t) \approx \sum_k s_{J,k}\phi_{J,k}(t) + \sum_k d_{J,k}\psi_{J,k}(t) + \sum_k d_{J-1,k}\psi_{J-1,k}(t) + \ldots\ldots + \sum_k d_{1,k}\psi_{1,k}(t)$$

where the $s_{J,k}$ and $d_{J,k}$ are wavelet transform coefficients, and $\phi_{J,k}$ and $\psi_{J,k}$ are approximating wavelet functions. The wavelet coefficients are given approximately by the integrals:

$$s_{J,k} \approx \int \phi_{J,k}(t) f(t) dt$$
$$d_{J,k} \approx \int \psi_{J,k}(t) f(t) dt$$

The functions

$$S_J(t) = \sum_k s_{J,k} \phi_{J,k}(t)$$
$$D_J(t) = \sum_k d_{J,k} \psi_{J,k}(t)$$

are called the smooth signals and the detail signals respectively. The orthogonal wavelet series approximation to a signal $f(t)$ is expressed as:

$$f(t) \approx S_J(t) + D_J(t) + D_{J-1}(t) + \ldots + D_1(t)$$

This constitutes a J-level wavelet decomposition of the signal.

4.3 Definition of wavelet based features

From a wavelet decomposition of a signal, we can compute the proportion of the energy that is included in the smooth and the details components. This gives a distribution that we can then use as a simple feature of the song. Given the nature of wavelet decomposition, this can be done locally for various portions of a song. For a j-level decomposition, represent the energy of each component as $E(D_1)...E(D_j)$ for the details and $E(S_j)$ for the smooth component. The total energy of a signal $f \in L^2(R)$ can be approximated as

$$T=\|f\|^2_{L^2(R)} \approx \sum_{j=1}^{J}\sum_{k=1}^{N/2^j}|d_{j,k}|^2+\sum_{k=1}^{N/2^J}|s_{J,k}|^2$$

One simple family of features we can look at to illustrate our proposed approach is based on the energy contained at the various levels of our decomposition. Define the relative energies:

$$E_{dj}=\frac{\sum_{k=1}^{N/2^j}|d_{j,k}|^2}{T} \qquad j=1,\ldots,J$$

$$E_{SJ}=\frac{\sum_{k=1}^{N/2^J}|s_{J,K}|^2}{T}$$

For a given choice of J and wavelet function $\psi()$, we decompose our signal into J level of signal and compute the following feature vector: $(E_{d1},E_{d2},...,E_{dJ},E_{sJ})$.

The objective in this work is to see if we can gather enough useful features from such decompositions to analyze and classify music signals. In the following, we use Daubechies' S8 wavelet, from the family of "symmlets" wavelets. For convenience, we adopt the notation $D_1 \ldots D_J\ S_J$ instead of $E_{d1},E_{d2},...,E_{dJ},E_{sJ}$ in the following sections.

4.4 Wavelet decomposition of signals

To illustrate the result from a wavelet decomposition, in Figure 4-2, we produce a signal called Doppler, by the following equation:

$$y_i = t(1-t)\sin(2.1\pi) / (t + 0.05)$$

where $t = i / n$, $i \in (1...n)$

The original signal is represented in the top box. The bottom box represents the smooth portion of the signal (S_4), in short, the low frequency components. Unlike the FFT (fast Fourier transform), we retain the time component of the signal. The other boxes (D_1-D_4) represent the details of the signal, D_1 corresponding to the highest frequencies. We see that most of the Gaussian noise is captured in D_1, while the higher frequency components of the signal proper are represented in D_2-D_4. The algorithm of producing Doppler signal and wavelet decomposition is shown in Appendix Source code2.

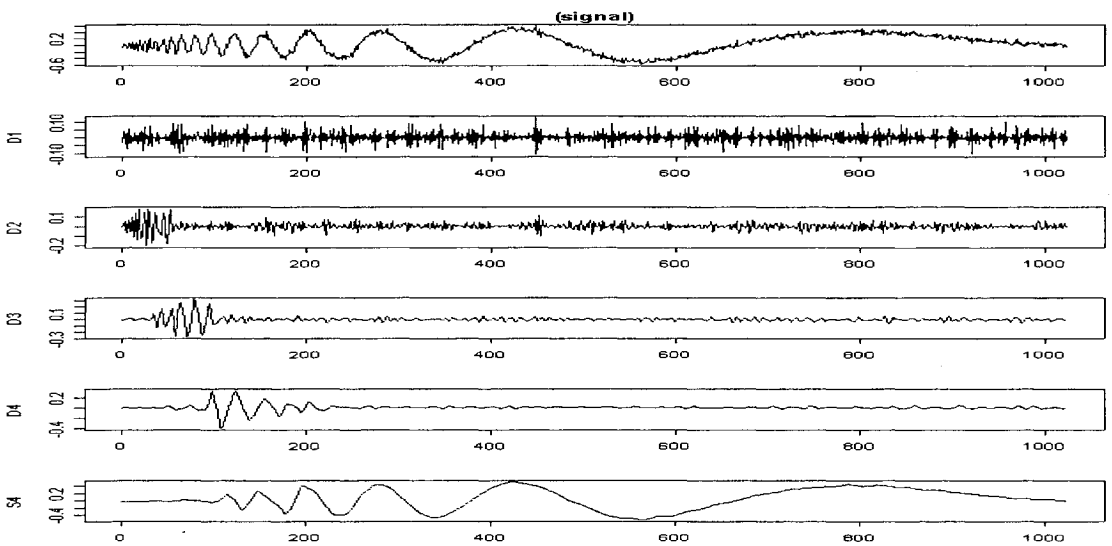


Figure 4-2 Decomposition of Doppler signal

In Figure 4-3, we present a two level decomposition of a one-minute clip from a song by the Chinese singer Zhang XinZhe. The bottom box (idwt) represents the signal reconstructed from the various parts of the wavelet decomposition. We compute correlation between original data and reconstructed data and it equals to 1. It shows the reconstructed data represents original data very well.

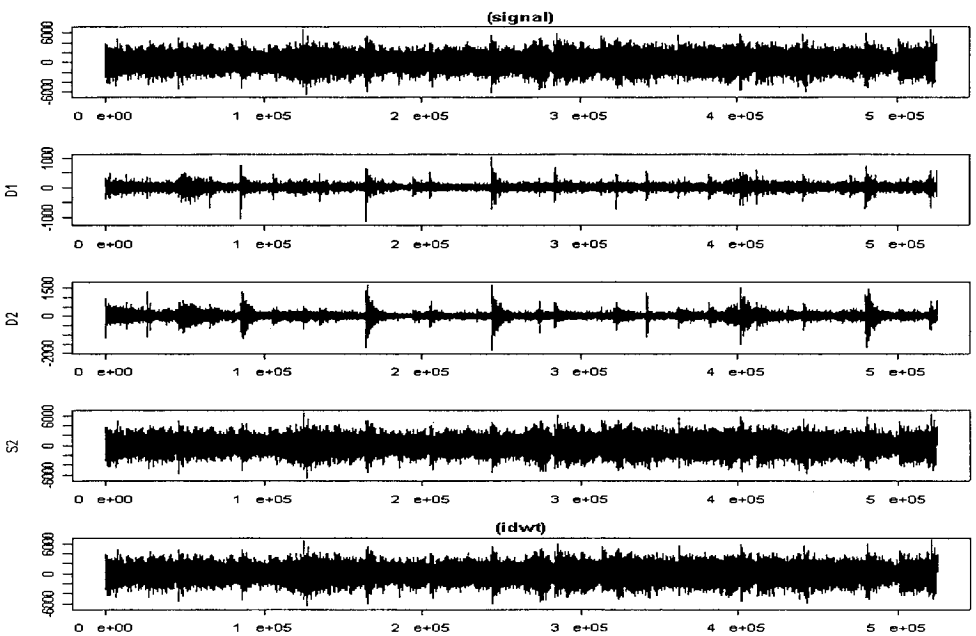


Figure 4-3 Two level decomposition of signal from the singer Zhang

Figure 4-4 provides a two level decomposition of a one-minute clip from a song by the French singer Edith Piaf. Signals D_1 and D_2 represent two levels of details and S represents the smooth low frequency signal. Signal idwt is the summation of signal D_1, $D2$, $S2$.

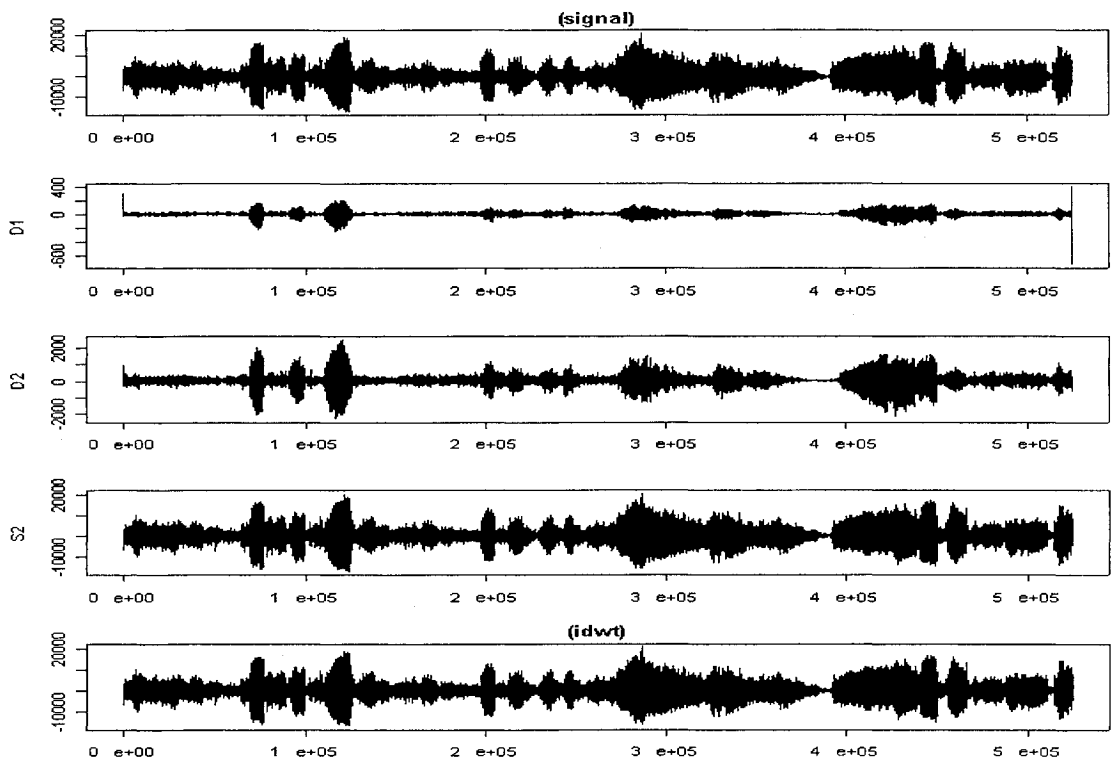


Figure 4-4 Two level decomposition of a signal from the singer Piaf

4.5 Computation of wavelet features

After defining the wavelet features, an algorithm was written in R for wavelet feature computation (See appendix 3 Source Code3). The result is stored in a matrix format. The following exploratory analysis is based on a two level decomposition.

4.6 Exploration of wavelet features

In order to determine the usefulness of our wavelet features for discriminating between singers, song types, etc., we conduct various exploratory analyses. Beginning with the descriptive box plots, we then make use of clustering techniques and analysis of variance.

Box Plots

Box plots are used in order to obtain an overview of the distribution of wavelet features. All of our box plots are obtained using two level wavelet decomposition. In Figure 4-5, we display box plots for the variable D_2 for songs from the singers Li Wen, Neil Diamond, Peter, Paul and Mary and Xu MeiJin. It can be seen that Neil Diamond has the largest median while Peter, Paul, Mary have the smallest. Neil Diamond and Xu MeiJin have a wider range on D_2 than Li Wen and Peter, Paul and Mary.

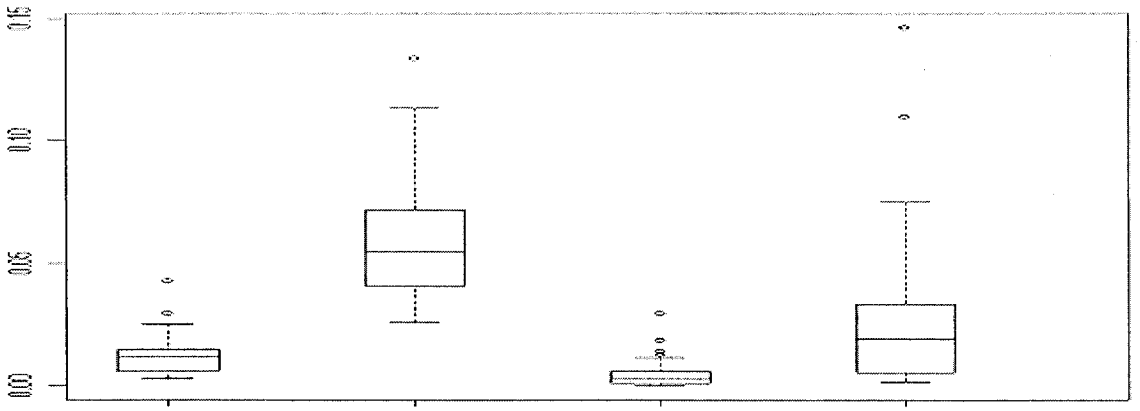


Figure 4-5 Box plot of D_2

In Figure 4-6 we compare Neil Diamond, Edith Piaf, Peter, Paul and Mary, Wang Fei, Xu MeiJin and Zhang XingZhe using the feature S_2. It can be seen that Piaf, known to have a low volume voice, is at the low end. The Box plots appear to distinguish among singers.

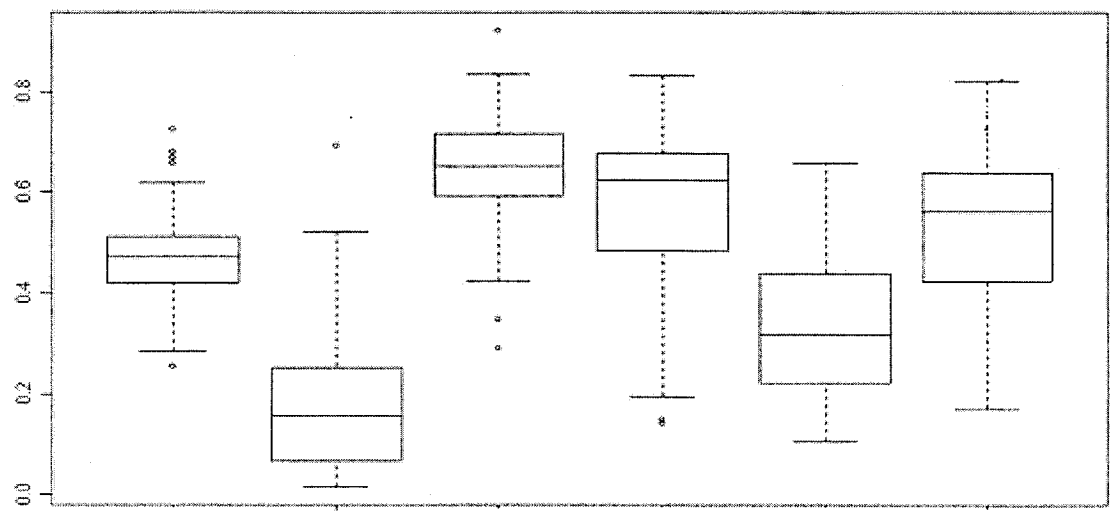


Figure 4-6 Box plot of S_2

Clustering

Cluster analysis is a statistical method used to group members that share certain properties. In Figure 4-7 we display the results of a hierarchical cluster analysis using features D_1 and D_2 only. We consider two songs each from two different singers. Two songs are from *Zhang* whereas the other two are from *Neil Diamond.* This is a very straightforward example, but we see that our features are good enough to group together songs and clips from the same singer.

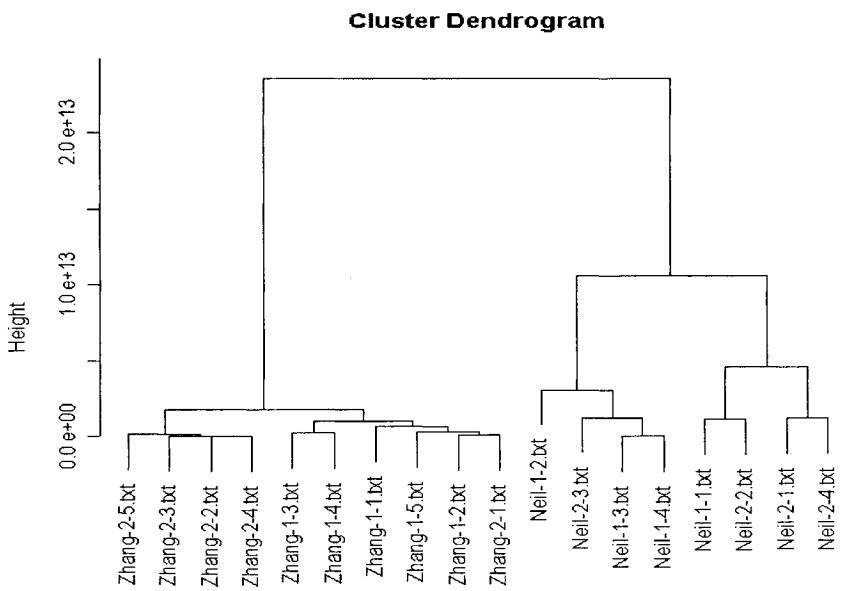


Figure 4-7 Clustering analysis of singers Zhang and Diamond

In Figure 4-8, we choose two songs from *Zhang* and two from *Piaf*. Once again, the singers are clearly distinguished.

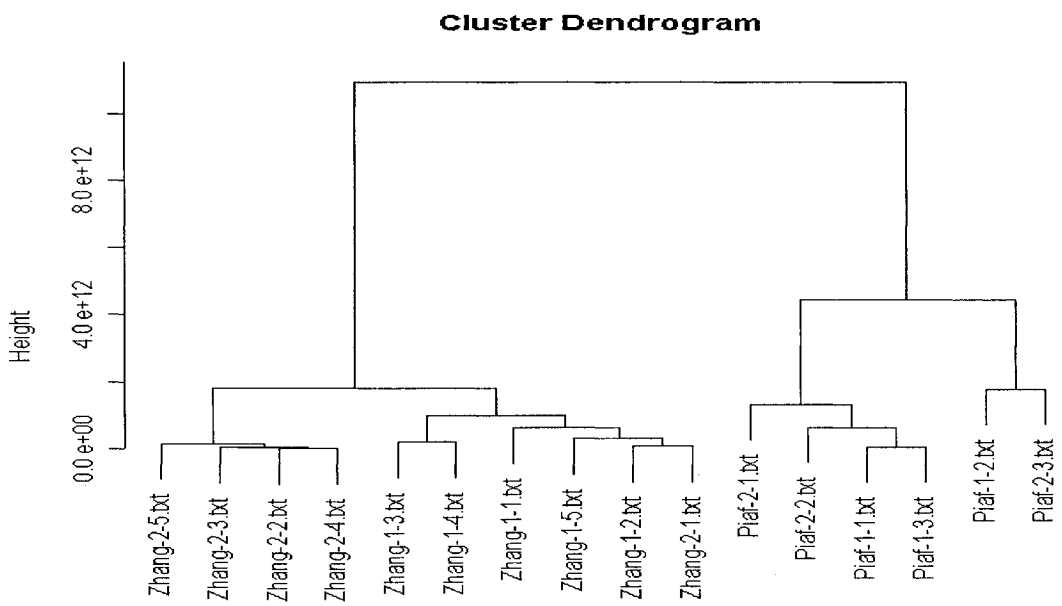


Figure 4-8 Clustering analysis of singers Zhang and Piaf

In Figure 4-9, we use 2-level wavelet decomposition on songs from two different singers. On the x-axis, we plot the proportion of energy of the signal included in D_1, while on the y-axis, we plot D_2. We then apply a K-means clustering with 2 clusters. The asterisks represent the centroids (centers of mass) of the two clusters. Boxes are drawn around points representing parts of songs from the two singers. This is again a simple case, but we see that we can get good discrimination even with only two simple features.

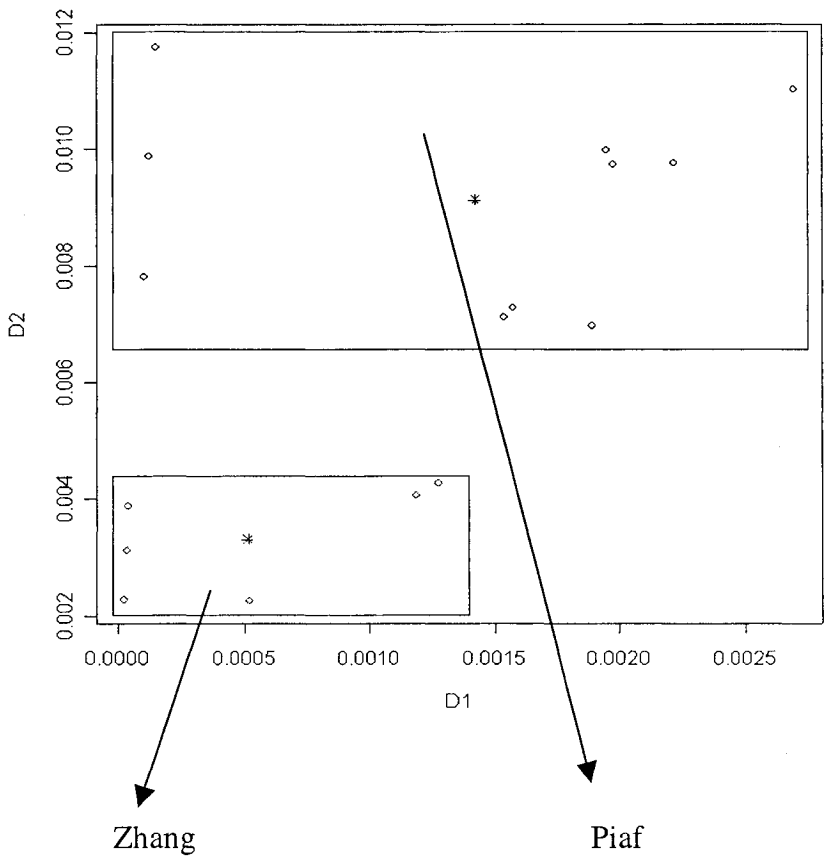


Figure 4-9 K-means analysis of singers Zhang and Piaf

In Figure 4-10, we compare *Domingo* and *Wang Fei*. It is interesting to note that the male singers appear to occupy a position and the lower quadrant compared to the female singers. This may be due to the fact that female singers have more energy in high frequencies than male singers.

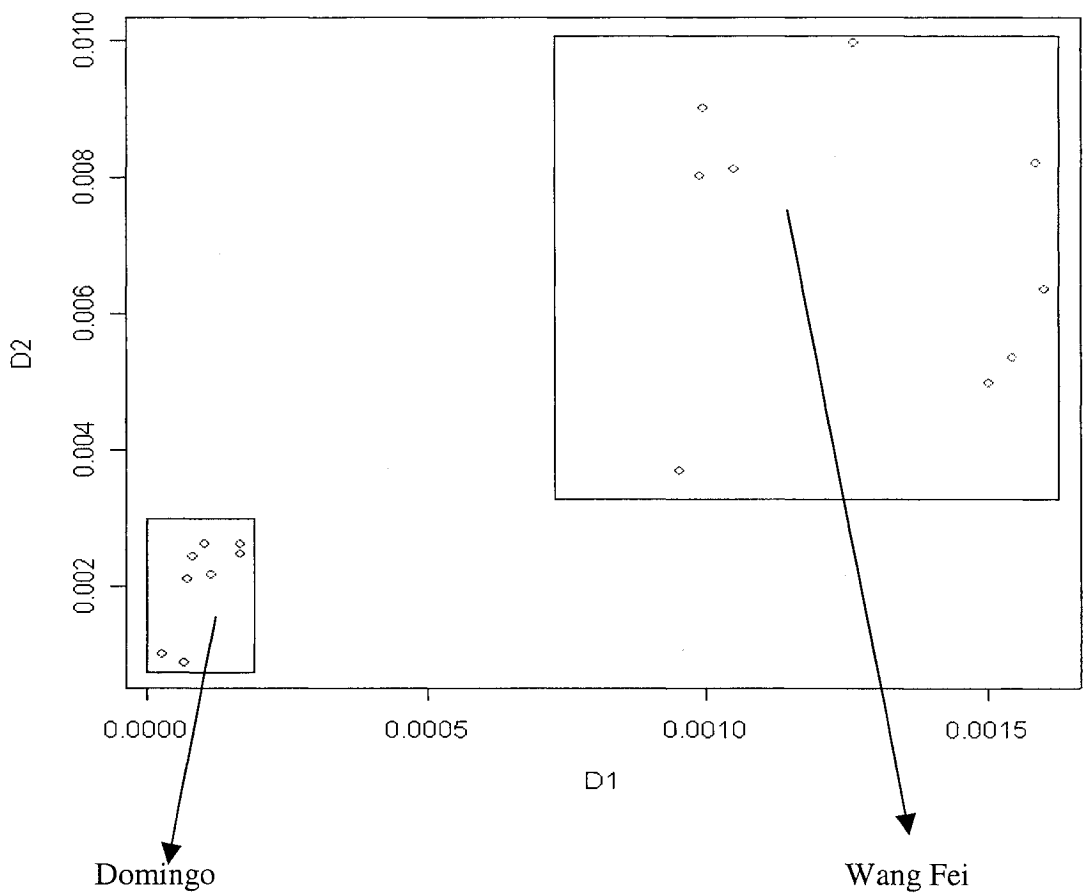


Figure 4-10 K-means analysis of singers Domingo and Wang Fei

In Figure 4-11, we exhibit the results for 13 singers using once again 2-level wavelet decompositions for D_1 and D_2. The list of singers is given in the Table 4-1. From Figure 4-11, we see that those two simple features are not enough to do advanced classification of the songs. We can however interpret some of the results; for example, *Holly Cole*, *Meng TingWei* and *WangFei*, three female pop singers, are in one tight cluster (4-8-12). Some results appear surprising as for example, *Placido Domingo*, *Harry Belafonte*, *Edith Piaf* in one group (2-3-10), and *Bob Marley, Li Wen, Jane Arden, Xu MeiJin* and *Lin YiLian* (1-5-6-7-13) in another group. We also see that Tina Turner is far away from all the other singers.

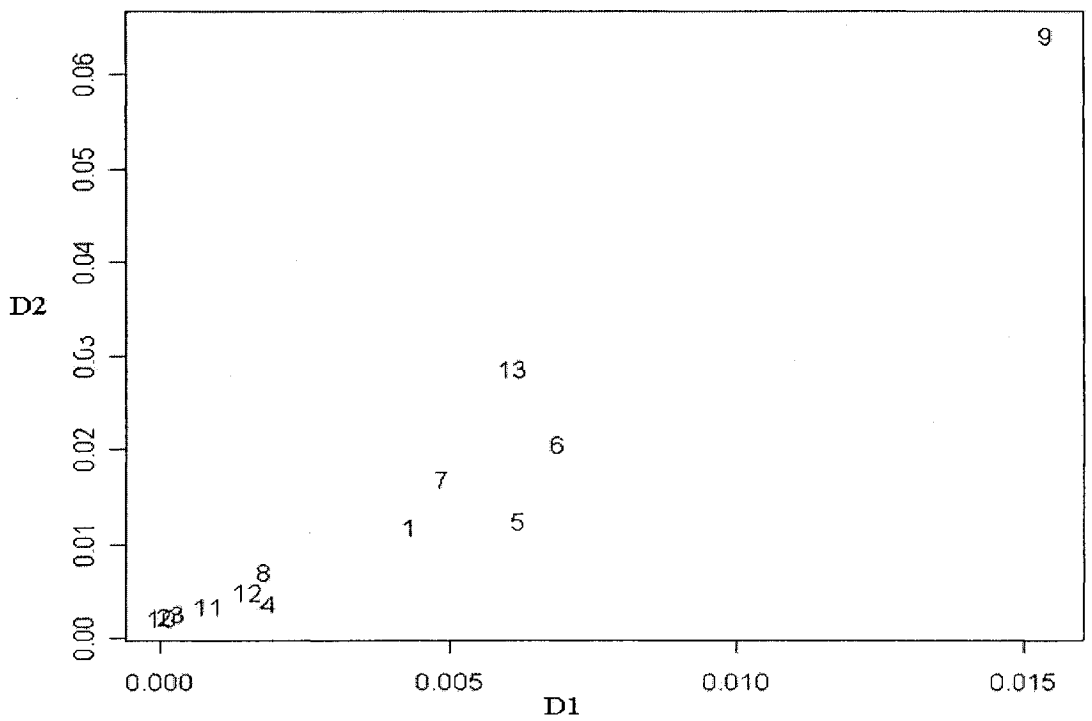


Figure 4-11 K-means display for 13 singers

Label	Singer's Name	Gender	Language	Type
1	Bob Marley	Male	English	Pop
2	Placido Domingo	Male	German	Classic
3	Harry Belafonte	Male	English	Pop
4	Holly Cole	Female	English	Pop
5	Jane Arden	Female	English	Pop
6	Lin YilLian	Female	Chinese	Pop
7	Li Wen	Female	Chinese	Pop
8	Meng TingWei	Female	Chinese	Pop
9	Tina Turner	Female	English	Pop
10	Edith Piaf	Female	French	Pop
11	Peter, Paul, Mary	Trio (M,M,F)	English	Country
12	Wang Fei	Female	Chinese	Pop
13	Xu MeiJin	Female	Chinese	Pop

Table 4-1 List of singers for Figure 4-11

Variation Analysis

In this section, an exploratory analysis is performed to test if certain Wavelet features can distinguish among different music categories. We select several songs as samples from each category. A nested model is postulated whereby different singers are nested within categories. A similar analysis could be carried out if one wishes to distinguish between singers. In order to reach the assumption of normality, we transformed the data by log. The model made use of features D_1, D_2 and T and is given as follows:

Model:

$$Y_{ij} = \mu_{..} + \alpha_i + \beta_{j(i)} + \varepsilon_{ij}$$

Y_{ij}: Value of one feature

α_i: Effect from music category

$\beta_{j(i)}$: Effect from songs

ε_{ij}: Error

Decision rule:

If the p-value of category < 0.10, the feature is considered significant and we conclude that there are differences among music categories with respect to that feature. In the following table, D_2P_1 represents feature D_2 of clip 1.

ANOVA Test for feature Log(D_2P_1)

General Linear Model: Log(D2_P1) versus Type, SingerName

```
Factor       Type Levels Values
Type        fixed      4 "C" "J" "L" "P"
Name(Type)  fixed      9 "AnneMurray"     "ShaniaTwain"     "DianaKrall"
                         "FrankSintras"   "LouisArmstrong"  "Carreras"
                         "Kanawa"         "CelineDion"      "JenniferLopez"

Analysis of Variance for Log(D2_P, using Adjusted SS for Tests

Source        DF     Seq SS     Adj SS     Adj MS       F      P
Type           3     3.1700     3.0933     1.0311    2.83  0.058
Name(Type)     5     2.0005     2.0005     0.4001    1.10  0.386
Error         26     9.4805     9.4805     0.3646
Total         34    14.6510
```

Table 4-2 ANOVA table of Log(D_2P_1)

Since the p-value of category equals 0.058, which is smaller than 0.10, we conclude that Log(D_2.P_1) can distinguish music categories.

From a plot of the residual of ANOVA model of the last test, the normality assumption for the data seems valid.

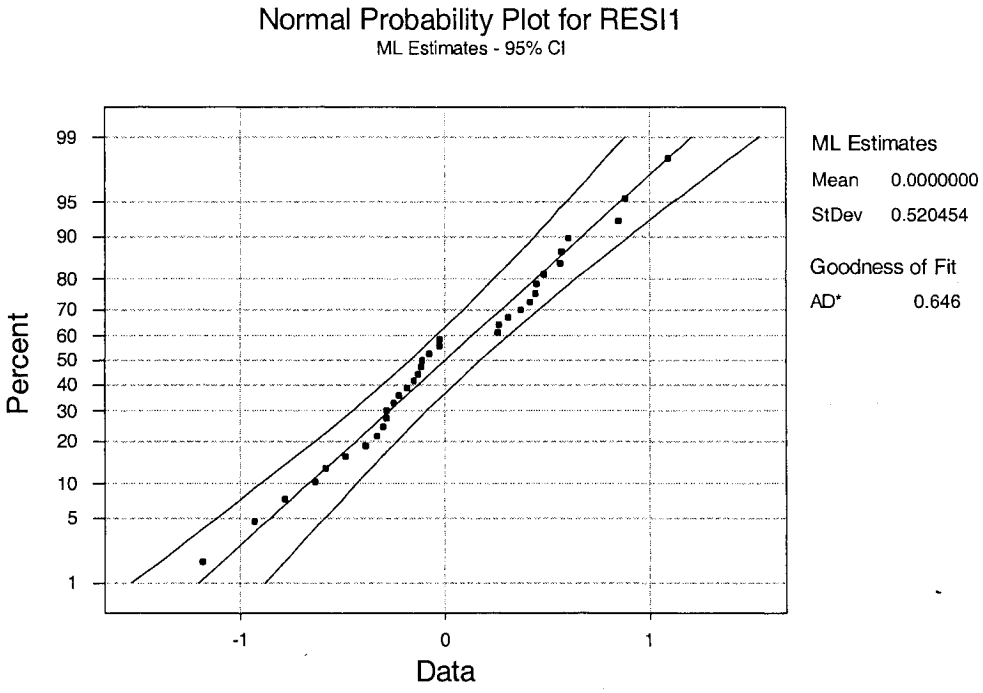


Figure 4-12 Residual Plot for ANOVA test for feature Log(D_2.P_1)

ANOVA Test for feature Log(D_1.P_1)

General Linear Model: Log(D_1_P_1)versus Type, SingerName

```
Factor        Type Levels Values
Type         fixed      4 "C" "J" "L" "P"
Name(Type)   fixed      9 "AnneMurray"      "ShaniaTwain"     "DianaKrall"
                          "FrankSintras"    "LouisArmstrong"  "Carreras"
                          "Kanawa"          "CelineDion"      "JenniferLopez"

Analysis of Variance for Log(D1_P, using Adjusted SS for Tests

Source        DF     Seq SS     Adj SS     Adj MS       F      P
Type           3     8.9239     9.1403     3.0468    6.51  0.002
Name(Type)     5     5.4807     5.4807     1.0961    2.34  0.070
Error         26    12.1755    12.1755     0.4683
Total         34    26.5801
```

Table 4-3 ANOVA table of Log(D_1.P_1)

We conclude that Log($D_1.P_1$) can distinguish music categories for the p-value equals to 0.002. From a plot of the residual of ANOVA model of the last test, the normality assumption for the data seems valid.

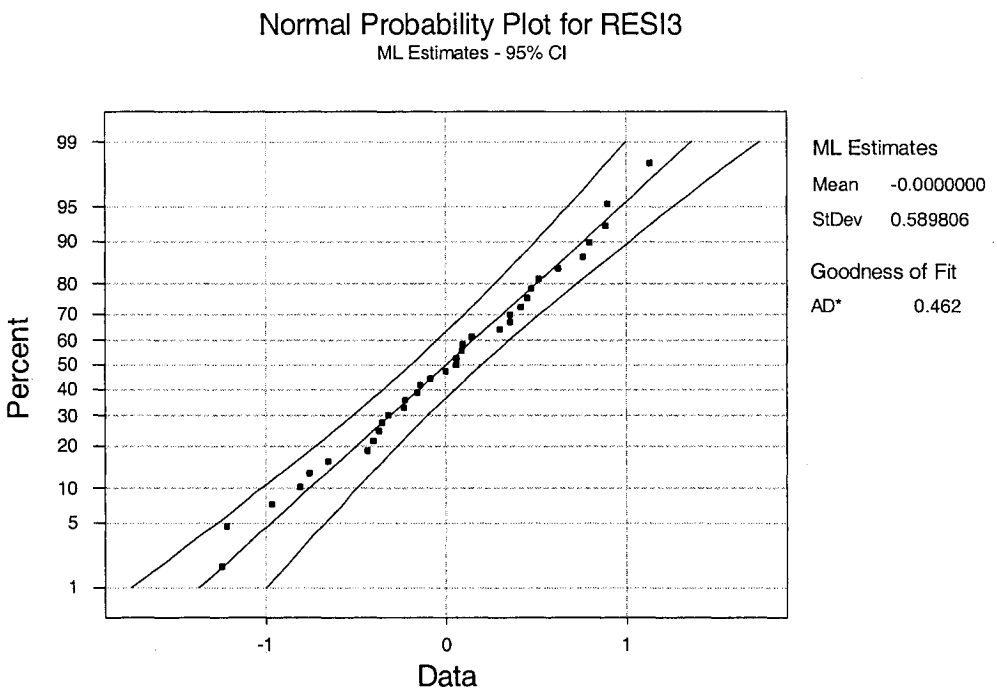
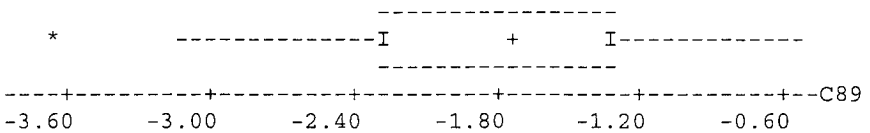


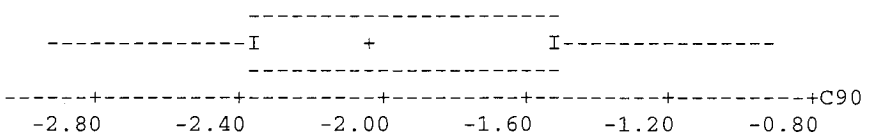
Figure 4-13 Residual Plot for ANOVA test for feature Log($D_1.P_1$)

Boxplot of Log(D1.P1)

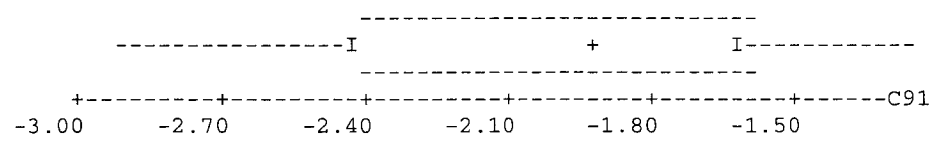
Pop



Country



Jazz



Opera

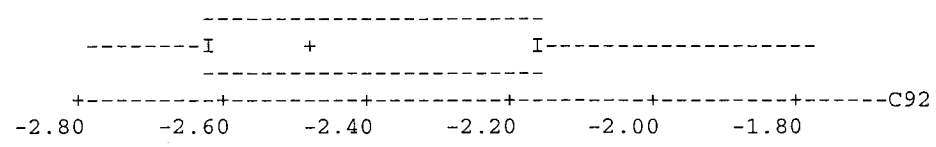


Figure 4-14Box plot of feature Log(D_2.P_1)

4.7 Conclusion of wavelet features

The ANOVA test implies that there are at least two music types that differ with respect to the mean of features D_1 and D_2.

Features D_1, D_2 are reasonable separators for some specific singers by K-means analysis and Clustering analysis.

The wavelet features only represent information on the energy contained in the voice signal. We will make use of these features in conjunction with others in a later chapter when we consider classification methods.

Chapter 5 Time Based Features

In this chapter we define a number of features that are functions of the wave signal as it evolves over time. These are different from the features that measured the energy, which were considered in the previous chapter.

5.1 Definition of time based features

We will introduce two families of features. One measures the time lag between crossings for a single profile of a wave (shown in Figure 5-1) and the other measures the time lag between two consecutive profiles (shown in Figure 5-2).

With reference to Figure 5-1, let I_{25}, I_{50}, I_{75} represent the time duration at 25, 50 and 75 percent respectively of the maximum energy. With reference to Figure 5-2, let B_{25} B_{50} B_{25} represent the time duration between consecutive peaks which have reached respectively 25, 50 and 75 percent of the maximum energy. The statistics used consist of the mean and variance of these variables calculated over all song clips within a category.

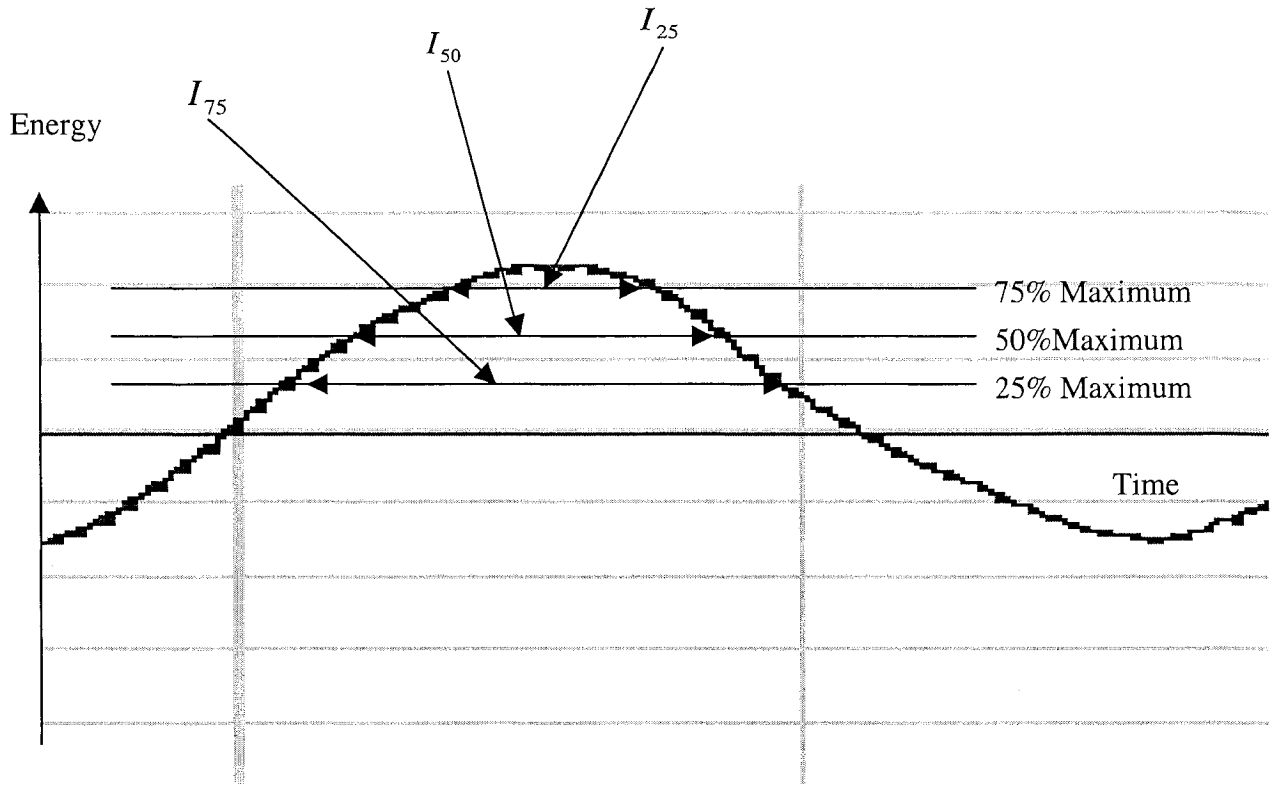


Figure 5-1 Definition of first group of time based features

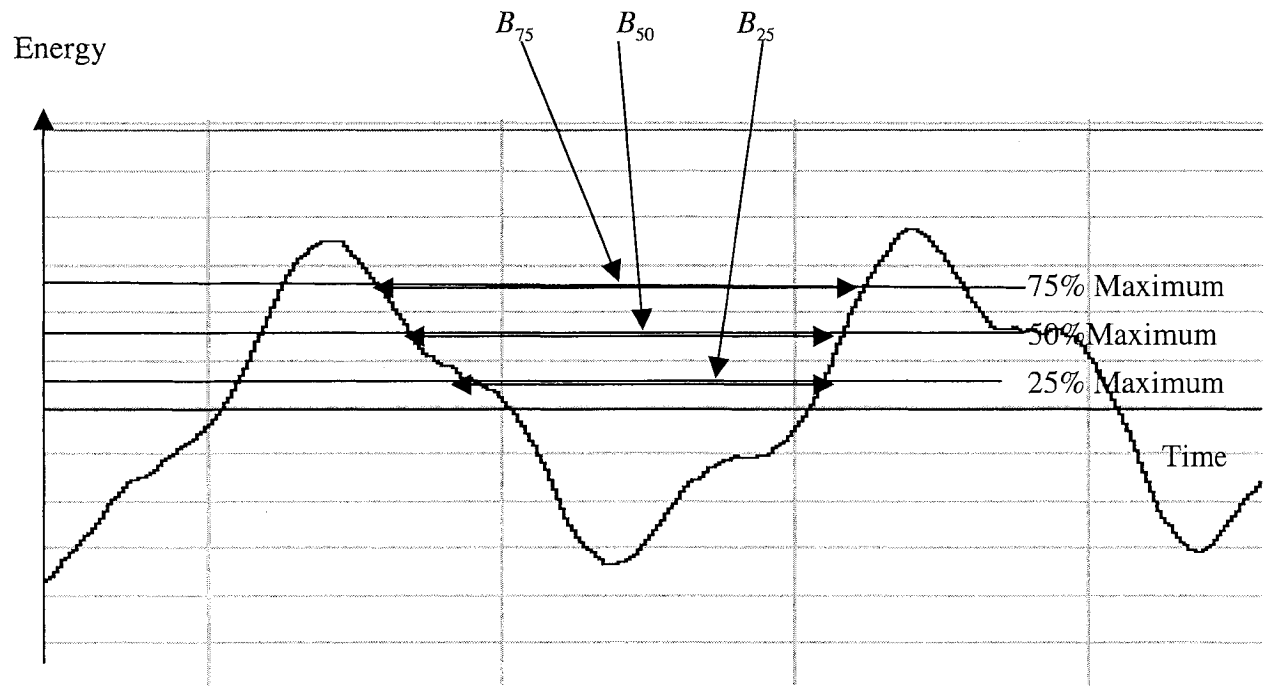


Figure 5-2 Definition of second group of time based features

5.2 Computation of time based features

An R program was written to compute time-based features (Shown in Appendix3 Source code4: Time-based feature1 algorithm, and code5: Time-based feature2 algorithm) The average and standard deviation of I_{25}, I_{50}, I_{75} B_{25}, B_{50}, B_{75} are computed for each clip and stored in a matrix shown in Table 5-1. They are denoted respectively by $\overline{B_{25}}$, SD(B_{25}). The first column represents the name of the respective clips.

Feature / Clips	$\overline{B_{25}}$	SD(B_{25})	$\overline{B_{50}}$	……	$\overline{I_{75}}$	$SD(I_{75})$
Clip1						
Clip2						
……						

Table 5-1 Computation of time based features

5.3 Exploration of time based features

An analysis of the time-based features was performed similar to what was done for the wavelet features. Specifically, a box plot over the music types and a nested model was postulated for each variable with the following results. We transformed the features by log to satisfy the assumption of normality by ANOVA model.

Log($\overline{B_{75}}$.P1)

In Figure 5-3, a Box Plot is presented for the mean of the log of B_{75} in Clip1, denoted as Log($\overline{B_{75}}$.P1), taken over the different music categories. It can be seen that there is little discerning power among categories. This is confirmed by the subsequent ANOVA test.

Boxplot of Log($\overline{B_{75}}$.P1)

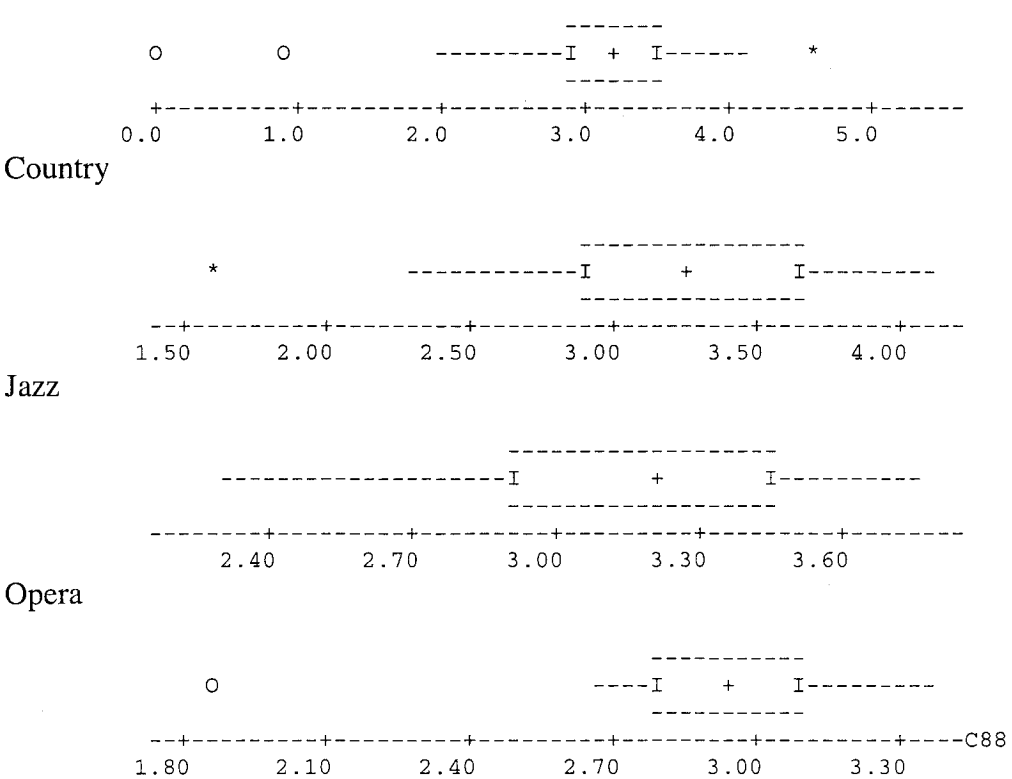


Figure 5-3 Box Plot of Log($\overline{B_{75}}$.P1)

General Linear Model: Log(Mean(B_{75})_P_1) versus Type, Name

```
Factor       Type Levels Values
Type         fixed      4 "C" "J" "L" "P"
Name(Type)   fixed      8 "AlanJackson"          "AnneMurray"
                          "DianaKrall"           "LouisArmstrong"
                          "Carreras"             "Kanawa"
                          "CelineDion"           "JenniferLopez"

Analysis of Variance for Log(Mean, using Adjusted SS for Tests

Source        DF     Seq SS     Adj SS     Adj MS       F      P
Type           3     1.3702     1.2816     0.4272    2.04  0.137
Name(Type)     4     0.7621     0.7621     0.1905    0.91  0.475
Error         22     4.5998     4.5998     0.2091
Total         29     6.7321
```

Table 5-2 ANOVA table of Log($\overline{B_{75}}$.P_1)

As the p-value of music type equals 0.137, Log($\overline{B_{75}}$.P_1) is not considered to be significant to distinguish music types.

The residuals of ANOVA model of Log($\overline{B_{75}}$.P_1) are close to a normal distribution as seen from the following plot.

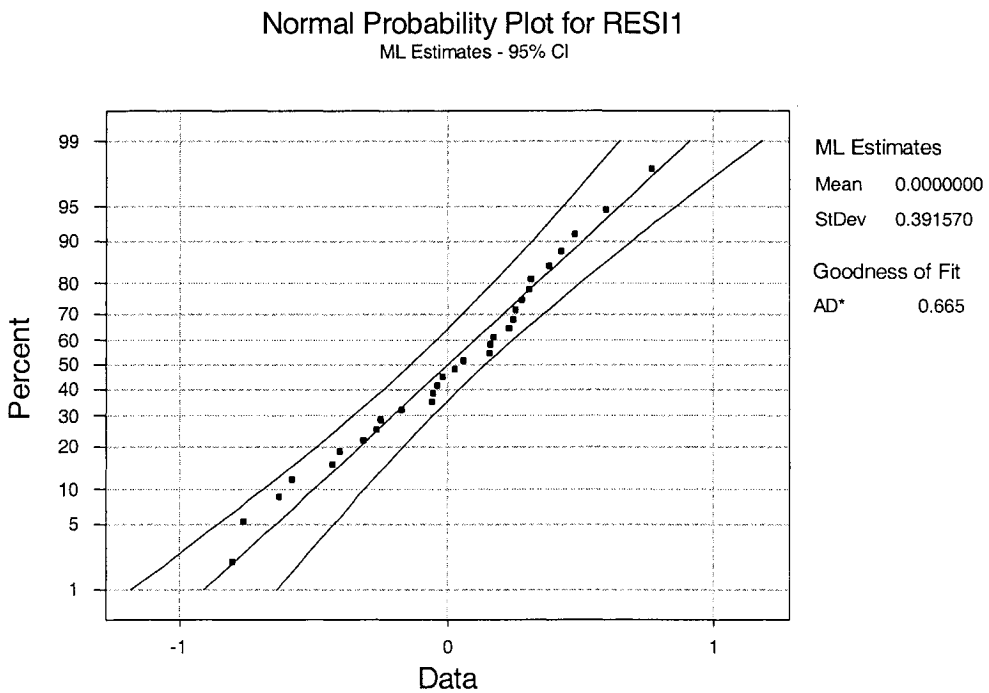


Figure 5-4 Residual Plot for ANOVA test for feature Log($\overline{B_{75}}$.P_1)

Log(SD (B_{75}).P_1)

A similar analysis is done for the standard deviation, denoted as Log(SD (B_{75}).P_1) in this example, which appears to be better at discriminating among categories.

General Linear Model: Log(SD(B75)_P1) versus Type, Name

```
Factor       Type Levels Values
Type        fixed      4 "C" "J" "L" "P"
Name(Type)  fixed      8 "AlanJackson"          "AnneMurray"
                       "DianaKrall"           "FrankSintras"
                       "Carreras"             "Kanawa"
                       "CelineDion"           "JenniferLopez"

Analysis of Variance for Log(SD(B, using Adjusted SS for Tests

Source        DF     Seq SS     Adj SS     Adj MS       F      P
Type           3     2.6017     2.4374     0.8125    2.45  0.090
Name(Type)     4     2.8653     2.8653     0.7163    2.16  0.107
Error         22     7.2960     7.2960     0.3316
Total         29    12.7630
```

Table 5-3 ANOVA table of Log(SD[B_{75}].P_1)

By the normal plot below, the residuals are close to a normal distribution.

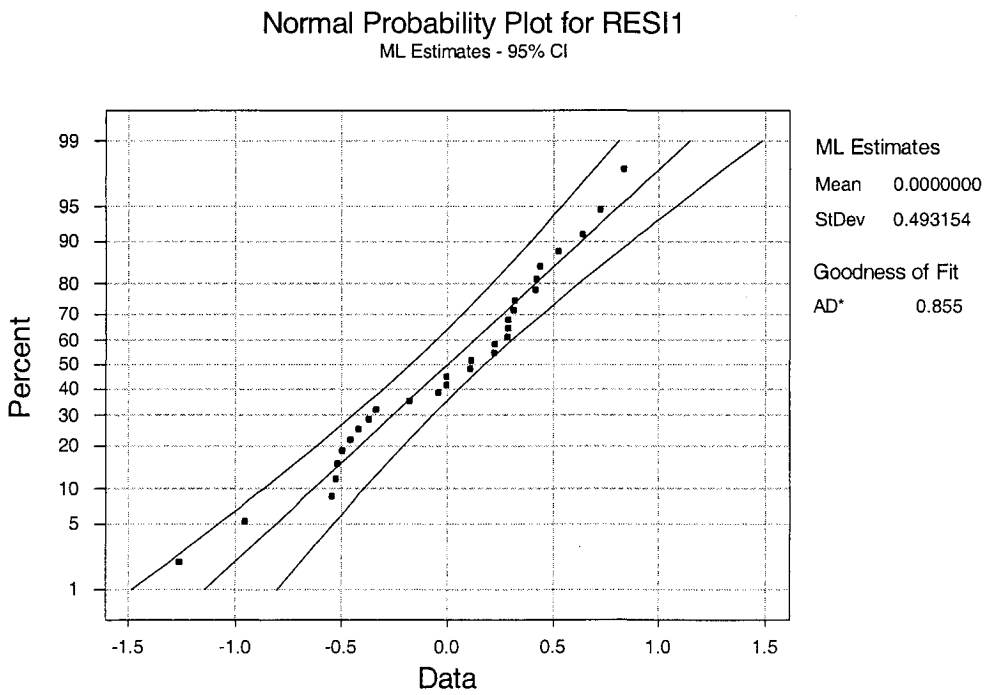


Figure 5-5 Residual Plot for ANOVA test for feature Log(SD[B_{75}].P_1)

Since the p-value of type is 0.090, we consider Log(SD(B_{75}).P_1) is significant to distinguish among music categories.

Log(SD(B_{50}).P_1)

General Linear Model: Log(SD(B_{50})_P_1) versus Type, Name

```
Factor       Type Levels Values
Type         fixed      4 "C" "J" "L" "P"
Name(Type)   fixed      8 "AlanJackson"          "AnneMurray"
                          "DianaKrall"           "FrankSintras"
                          "Carreras"             "Kanawa"
                          "CelineDion"           "JenniferLopez"

Analysis of Variance for Log(SD(B, using Adjusted SS for Tests

Source        DF     Seq SS     Adj SS     Adj MS       F      P
Type           3     1.8005     1.7491     0.5830    3.49  0.033
Name(Type)     4     2.1234     2.1234     0.5309    3.18  0.033
Error         22     3.6760     3.6760     0.1671
Total         29     7.5999
```

Table 5-4 ANOVA table of Log(SD[B_{50}].P_1)

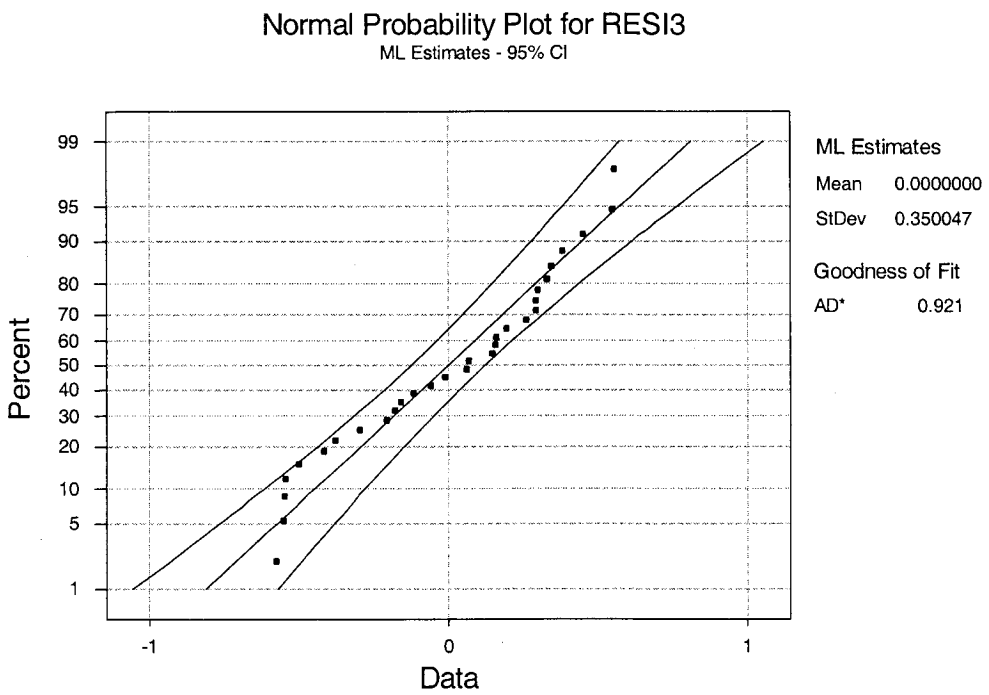


Figure 5-6 Residual Plot for ANOVA test for feature Log(SD[B_{50}].P_1)

Since the p-value of type is equal to 0.033, we consider Log(SD(B_{50})_P_1) is significant to distinguish music categories. The residuals appear to be normally distributed as seen in Figure 5-6.

$Log(SD(I_{50}).P_1)$

General Linear Model: $LOG(SD(I_{50})\_P_1)$ versus Type, Name

```
Factor       Type Levels Values
Type        fixed      4 "C" "J" "L" "P"
Name(Type)  fixed      8 "AlanJackson"          "AnneMurray"
                       "DianaKrall"           "FrankSintras"
                       "Carreras"             "Kanawa"
                       "CelineDion"           "JenniferLopez"

Analysis of Variance for LOG(SD(I, using Adjusted SS for Tests

Source        DF     Seq SS     Adj SS     Adj MS       F      P
Type           3    0.81628    0.69933    0.23311    2.81  0.063
Name(Type)     4    0.36484    0.36484    0.09121    1.10  0.382
Error         22    1.82714    1.82714    0.08305
Total         29    3.00826
```

Table 5-5 ANOVA table of $Log(SD[I_{50}].P_1)$

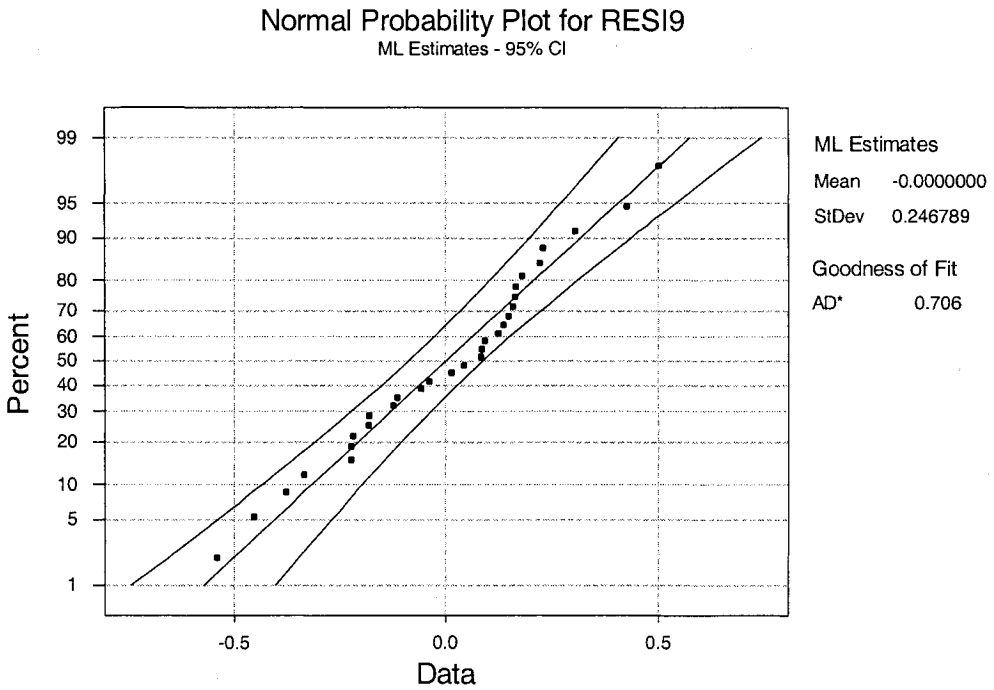


Figure 5-7 Residual Plot for ANOVA test for feature $Log(SD[I_{50}].P_1)$

Since the p-value of type is 0.063, we consider $Log(SD(I_{50})\_P_1)$ is significant to distinguish among music categories. The plot of residuals in Figure 5-7 supports the normality assumption.

$Log(SD(I_{25}).P_1)$

General Linear Model: $LOG(SD(I_{25})\_P_1)$ versus Type, Name

```
Factor        Type Levels Values
Type         fixed      4 "C" "J" "L" "P"
Name(Type)   fixed      8 "AlanJackson"          "AnneMurray"
                          "DianaKrall"           "FrankSintras"
                          "Carreras"             "Kanawa"
                          "CelineDion"           "JenniferLopez"

Analysis of Variance for LOG(SD(I, using Adjusted SS for Tests

Source        DF     Seq SS     Adj SS     Adj MS       F      P
Type           3    1.24928    1.01678    0.33893    3.70  0.027
Name(Type)     4    0.75563    0.75563    0.18891    2.06  0.120
Error         22    2.01522    2.01522    0.09160
Total         29    4.02013
```

Table 5-6 ANOVA table of $Log(SD[I_{25}].P_1)$

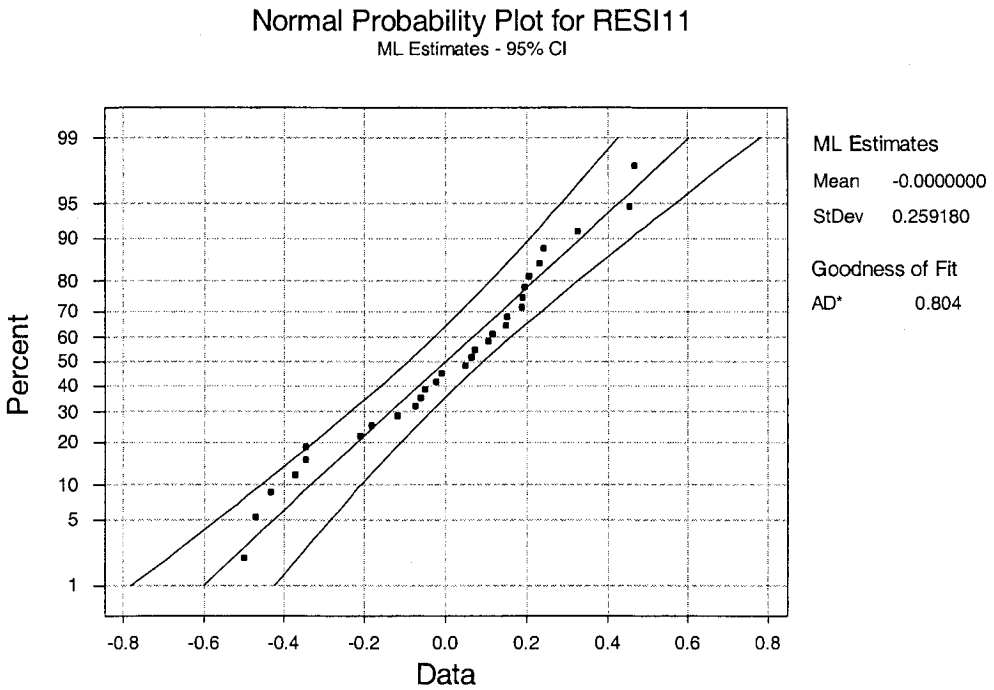


Figure 5-8 Residual Plot for ANOVA test for feature $Log(SD[I_{25}].P_1)$

Since the p-value of type is equal to 0.027, we consider $Log(SD(I_{25})\_P_1)$ is significant to music categories. The residuals appear normal as seen in Figure 5-8.

5.4 Conclusion of time based features

The ANOVA test implies that at least two music types differ with respect to the standard deviation of I_{50}, I_{25}, B_{75} and B_{50}. Since the voice signal is variable, the standard deviation features represent better characteristics than the mean of the features.

By the preliminary statistical tests determined in Chapter 4 and 5, there were four significant features: D_1, D_2, I_{50}, I_{25}, B_{75} and B_{50}. At this point, we are not sure if those features are able to distinguish among music types. In the following chapter, we classify music types with those selected features as well as all of the available features.

Chapter 6 Classification

In this chapter, linear discriminant analysis(LDA), quadratic discriminant analysis(QDA) and tree-based methods are used for classifying songs into four categories. In addition, an adaptive tree approach is developed to compare with the above three methods.

6.1 Discriminant analysis

Discriminant analysis is a statistical method used for classifying an observation into one of several populations. For example, in plant taxonomy a botanist may wish to classify a new specimen as one of several recognized species of a flower. In routine banking or commercial finance an officer or analyst may wish to classify loan applicants as low or high credit risks on the basis of the elements of certain accounting statements. In each case the decision maker wishes to classify from simple functions of the observation vector rather than complicated regions in the higher-dimensional space of the original vector.

LDA is a classification method that discriminates between groups on the basis of a linear function of the data. The linear discriminate function is determined from samples collected for each group.

For example: in the case of 2 populations with common covariance, the linear discriminate function is:

$$W_{12} = (\bar{X}_1 - \bar{X}_2)' S^{-1} X - \frac{1}{2}(\bar{X}_1 - \bar{X}_2)' S^{-1}(\bar{X}_1 + \bar{X}_2)$$

where: $\bar{X}_1$: Sample mean of population 1

$\bar{X}_2$: Sample mean of population 2

X : New observation, to be classified into either population

S : Sample covariance matrix

Decision rule: If $W_{12}>0$, the new observation is classified into Population1; otherwise it is classified into Population2.

While LDA uses a linear discriminant function, QDA uses a quadratic function to allocate new observations. In two-dimensional space, such a rule will define a curve rather than a line.

6.2 Tree classification

Classification trees are used to predict membership of cases or objects in the classes of a categorical dependent variable from their measurements on one or more predictor variables. Classification tree analysis is one of the main techniques used in Data Mining. The goal of Classification trees is to predict or explain responses on a categorical dependent variable, and as such, the available techniques have much in common with the techniques used in the more traditional methods of Discriminant Analysis, Cluster Analysis, Nonparametric Statistics, and Nonlinear Estimation. The flexibility of classification trees make them a very attractive analysis option, but this is not to say that their use is recommended to the exclusion of more traditional methods. Indeed, when the typically more stringent theoretical and distributional assumptions of more traditional methods are met, the traditional methods may be preferable. But as an exploratory technique, or as a technique of last resort when traditional methods fail, classification trees are, in the opinion of many researchers, unsurpassed[4].

6.3 Dataset definitions and computation

We define a matrix that contains the following information: category, gender, singer's name, decomposition information and the features. We build the dataset using the features developed in Chapter 4 and 5. For each song, we consider the 15 features from each of the 5 clips, thus producing a 75-long feature vector. These features are: $B_{25}.P_1$, $B_{50}.P_1$, $B_{75}.P_1$,... ..., $I_{25}.P_5$, $I_{50}.P_5$, $I_{75}.P_5$, where P_1-P_5 refer to the clips. The beginning 6 columns of the format represents information of the song in this row. Finally, the resulting matrix contains 81 columns and 130 rows. The details of the dataset are shown in following table.

	Data[,1]	Data[,2]	Data[,3].....	....Data[,81]
Song1				
Song2				
.....				
Song129				
Song130				

Table 6-1 Definition of dataset

The definition of the dataset is shown below in more detail:

Column :

Data[1,2,3,4,5,6]: Number, Type, Gender, Language, Name of Singer, Name of Song

Clip1

Data[7,8,9]: Wavelet decomposition features D_1 , D_2 , T of Clip1

Data[10,11,12,13,14,15]: $\overline{B_{75}}$, SD(B_{75}), ($\overline{B_{50}}$), SD(B_{50}), $\overline{B_{25}}$, SD(B_{25}) of Clip1

Data[16,17,18,19,20,21]: $\overline{I_{75}}$, SD(I_{75}), $\overline{I_{50}}$, SD(I_{50}), $\overline{I_{25}}$, SD(I_{25}) of Clip1

Columns from 22 to 81 are the same as column 7-21 from clips 2 to 5 respectively.

6.4 Classification results

In this section, classification results are presented first in terms of the features selected in previous chapters and then using all of the features together. In both instances we will make use of LDA, QDA and the Tree method. Finally, an adaptive tree approach is used for classification. A Cross validation method will be used to quantify the success of a classification method. In our context this means that all the songs but one will be used to calibrate the method, which is then tested on the song left out. This procedure is repeated for each song. The proportion of songs correctly classified provides a measure of the success of the classification.

6.4.1 Classification using selected features

In chapters 4 and 5, the features D_1, D_2, standard deviation of I_{50}, I_{25}, B_{75} and B_{50} were considered to be significant. We applied those selected features in LDA, QDA and the Tree method and obtained the results exhibited in Table 6-2 and Figure 6-1.

Discussion:

By using the LDA method, the proportion of correct classification using cross validation is only 0.50. For the QDA no results were obtained given the large correlation in the data. Figure 6-1 shows how many cuts the tree method used to reduce the error and how many leaves to reach the end. The tree method yielded a proportion correctly classified is equal to 0.53. These results are not satisfactory. Thus, in the following section, we will try to use all of the features to see if we can get a better classification result.

LDA

Summary of Classification with Cross-validation

Put into Group	....True Group.... "C"	"J"	"L"	"P"
"C"	3	4	0	5
"J"	5	7	5	2
"L"	1	4	7	0
"P"	6	1	0	16
Total N	15	16	12	23
N Correct	3	7	7	16
Proportion	0.200	0.438	0.583	0.696

N = 66 N Correct = 33 Proportion Correct = 0.500

Table 6-2 LDA using selected features

Tree method

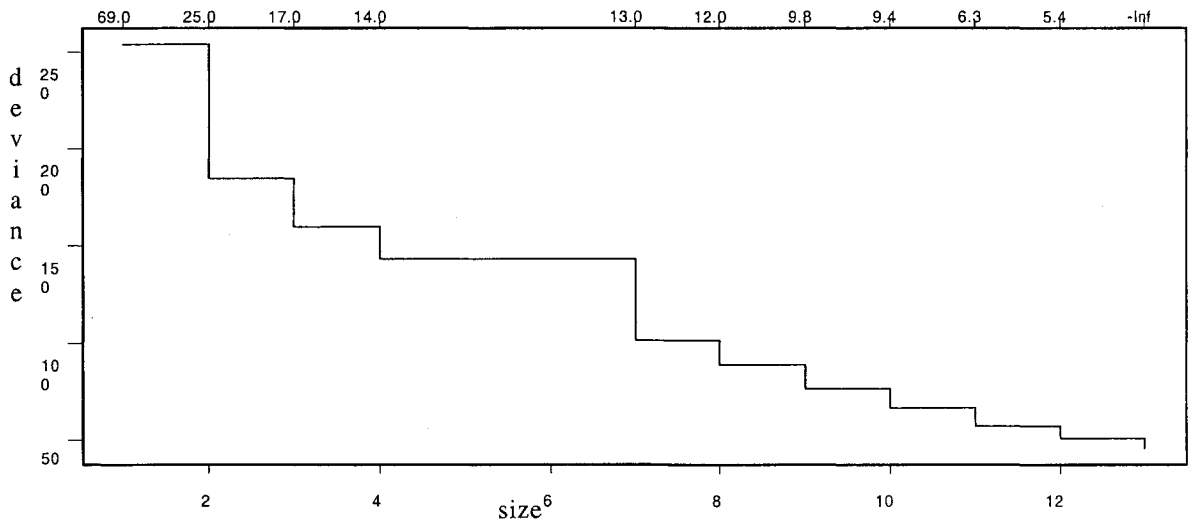
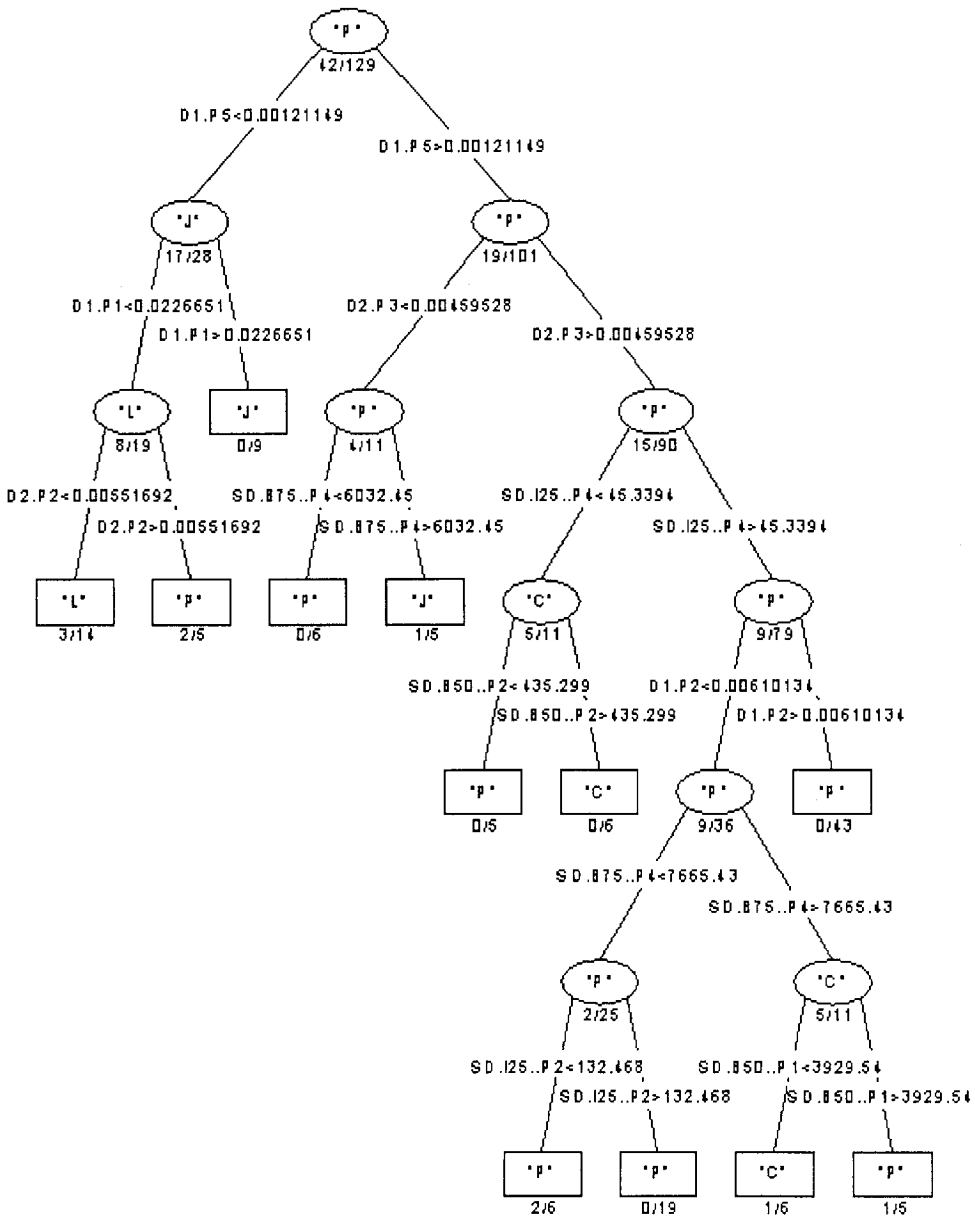


Figure 6-1 Size of Tree using selected features

The Tree method will try each predictor to separate data and find a best predictor first. Then find a second best predictor to separate until there is no predictor that can improve the proportion of classification. The Tree method needs to decide the size because not every predictor is useful for classification. The trees are grown using the deviance to decide on the splits, until terminal nodes are too small to split. As shown in Figure 6-1 and Figure 6-2, 11 cuts lead to the final 12 leaves in Tree classification. [8] [9]



P=Pop, C=Country, J=Jazz, L=Classic

Figure 6-2 Tree method using selected features(Pop(4 languages), Jazz, Country, Classic)

6.4.2 Classification using all the features

In this section, we incorporate all the features available in the classification methods. The four music categories will be classified under two conditions: 1) Pop songs in English; 2) Pop songs in English, French, Spanish and Chinese.

Classification using categories: Pop(4 languages), Country, Jazz, Classic

In this section, 87 Pop songs, 15 Country songs, 16 Jazz songs and 12 Classic songs are used for classification. In Pop songs, 23 songs are in English, 20 songs are in French, 22 songs are in Chinese and 22 songs are in French. Country and Jazz songs are in English and Classic songs are in Italian.

LDA

We use LDA to classify data by types, which contains all of the features for a song. Table 6-3 exhibits the confusion table. The proportion of correctly classified songs by cross validation is once again only 0.50.

```
Summary of Classification with Cross-validation
Put into      ....True Group....
Group           "C"       "J"       "L"       "P"
"C"               3         2         1        25
"J"               0        10         4         6
"L"               2         2         3         7
"P"              10         2         4        49
Total N          15        16        12        87
N Correct         3        10         3        49
Proportion    0.200     0.625     0.250     0.563

N =  130     N Correct =   65     Proportion Correct = 0.500
```

Table 6-3 LDA (Pop(4 languages), Country, Jazz, Classic)

Tree method

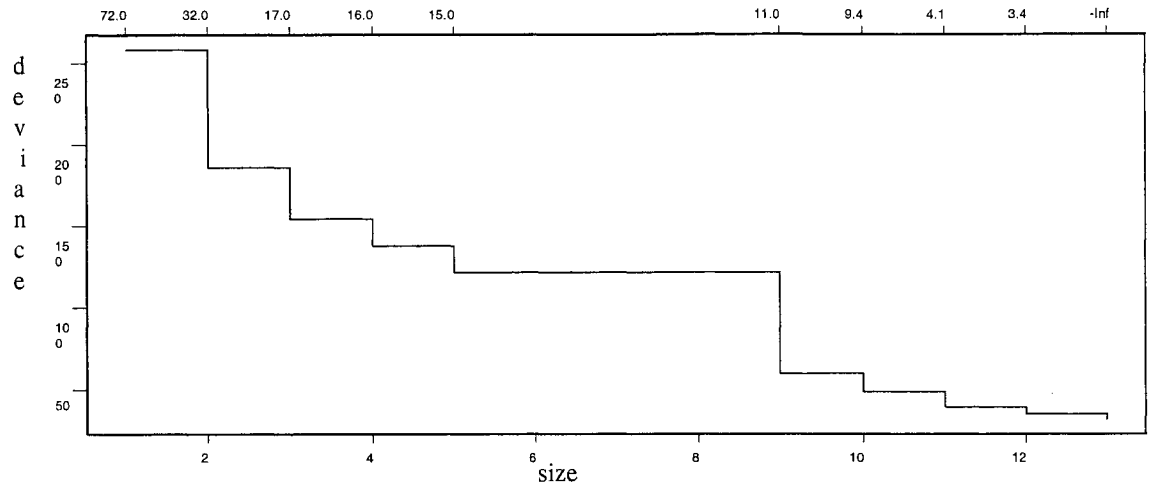
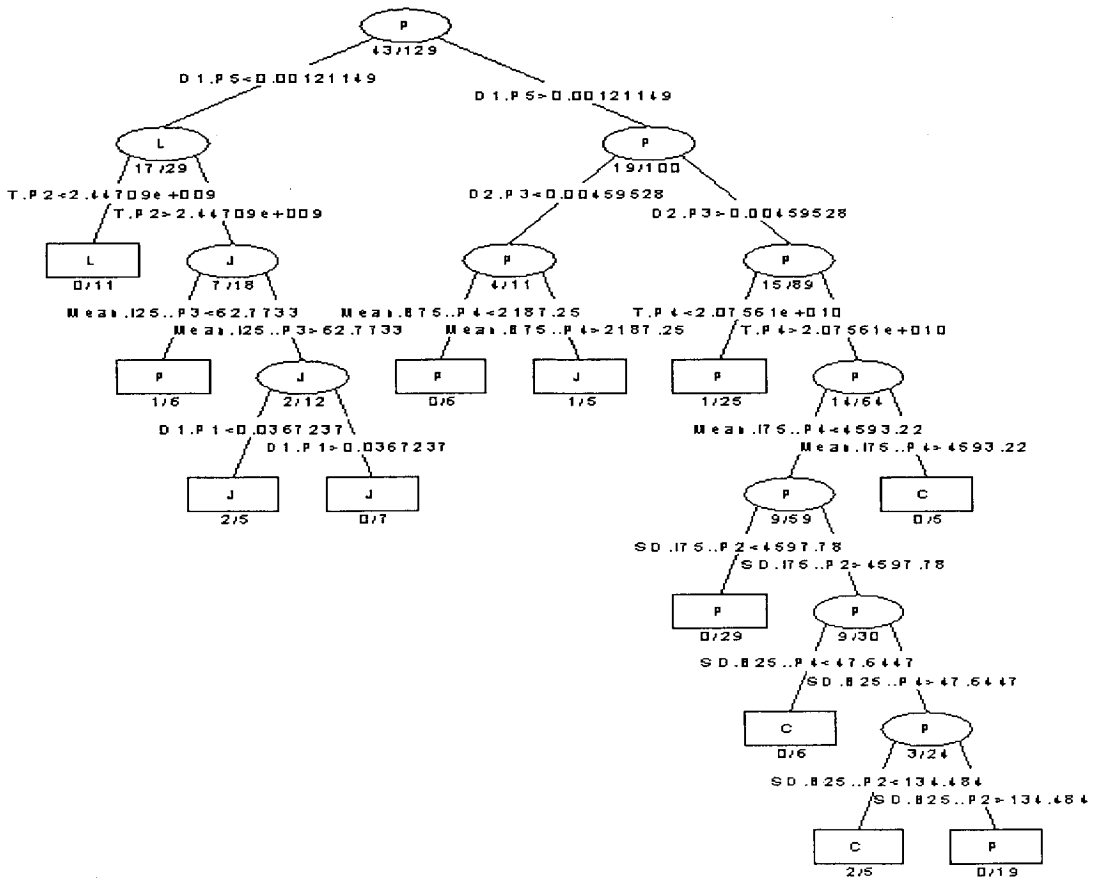


Figure 6-3 Size of Tree using all features



P=Pop, C=Country, J=Jazz, L=Classic

Figure 6-4 Tree method(Pop(4 languages), Country, Jazz, Classic)

The tree method yielded a proportion of correct classification equal 0.66 from Table 6-4. From Figure 6-4, we find that $D_1.P_5$, $T.P_2$, $D_2.P_3$, $\overline{I_{25}}.P_2$, $\overline{I_{25}}.P_2$ are important features for classification.

True Decision	Pop	Country	Jazz	Classic
Pop	0.67	0.21	0.12	0.0
Country	0.40	0.40	0.20	0.0
Jazz	0.12	0.12	0.76	0.0
Classic	0.0	0.0	0.18	0.82

Table 6-4 Confusion table of tree method (Pop(4 languages), Country, Jazz, Classic)

Discussion:

Among the classification methods considered, the tree method is best. It gives us a satisfactory proportion 0.66 of correct classification.

Table 6-4 is a confusion table based on the tree classification. The diagonal entries indicate the proportions of correctly classifying a song given its original class. The probabilities of correct classification for the four categories of music are as follows, (Pop 0.67; Country 0.40; Jazz 0.76; and Classic 0.82). The probability that a song will be recognized correctly is 0.66. Our method provides a satisfactory result except for the Country type.

The off diagonal entries represent the proportions of misclassification. It is evident that Pop, Jazz and Country songs are never misclassified as being of type Classic. There appears to be some imprecision between Country and Pop. These observations lead us to consider an adaptive tree method described in section 6.4.3.

Since 0.40 of Country songs are recognized as Pop, we might consider merging Country songs and Pop songs into one category. This will be discussed in a later section.

Classification using categories: Pop (English), Country, Jazz, Classic

In this section, 23 English Pop songs, 14 English Country songs, 16 English Jazz songs and 12 Italian Classic songs are used for classification. LDA gets a proportion of 0.561 and the Tree methods gets a proportion of 0.652.

LDA

```
Summary of Classification with Cross-validation

Put into      ....True Group....
Group             C        J        L        P
C                 5        1        0        8
J                 6       10        2        3
L                 0        5       10        0
P                 4        0        0       12
Total N          15       16       12       23
N Correct         5       10       10       12
Proportion    0.333    0.625    0.833    0.522

N =   66     N Correct =   37     Proportion Correct = 0.561
```

Table 6-5 LDA (Pop (English),Country, Jazz, Classic)

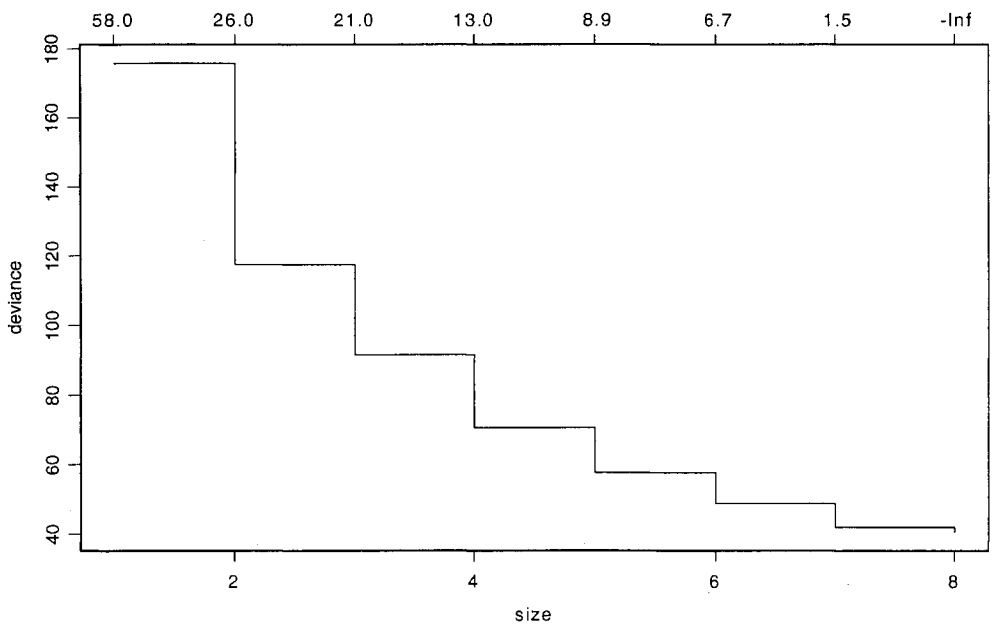
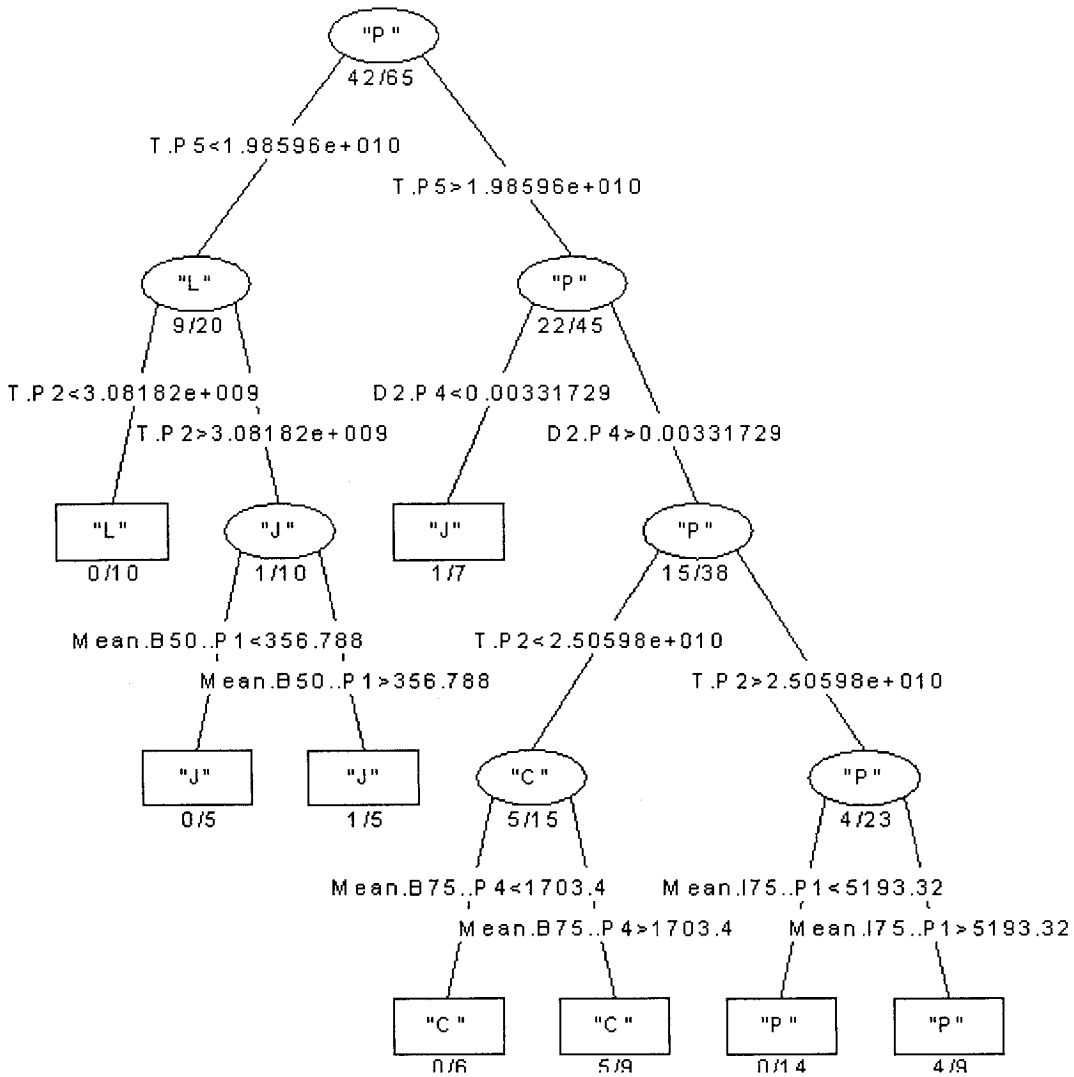


Figure 6-5 Size of Tree using Pop (English), Country, Jazz, Classic



P=Pop(English), C=Country, J=Jazz, L=Classic

Figure 6-6 Tree method (Pop(English),Country, Jazz, Classic)

True Decision	Pop	Country	Jazz	Classic
Pop	0.70	0.22	0.08	0.0
Country	0.43	0.43	0.14	0.0
Jazz	0.12	0.12	0.76	0.0
Classic	0.0	0.0	0.25	0.75

Table 6-6 Confusion table of tree method (Pop (English),Country, Jazz, Classic)

Discussion:

The Tree method is still best when compared to LDA. It gives a proportion of correct classification equal to 0.66 by the confusion table. Our model is doing well for Pop, Jazz and Classic types, but for the Country type the proportion is still low. Also, the confusion table is very similar to Table 6-4, showing that different languages do not affect the classification of music type. The Country type is recognized at a low proportion under both of the two conditions, but it has a high proportion to be recognized as Pop type. Thus, in the next section, we will consider 3 categories: Pop(English)/Country, Jazz and Classic .

Classification using 3 categories: Pop(English)/Country, Jazz, Classic

In this section, there are 37 songs of Pop(English)/Country type, 16 songs of Jazz type and 12 songs of Classic type that are used for classification. We report results using only the tree method here.

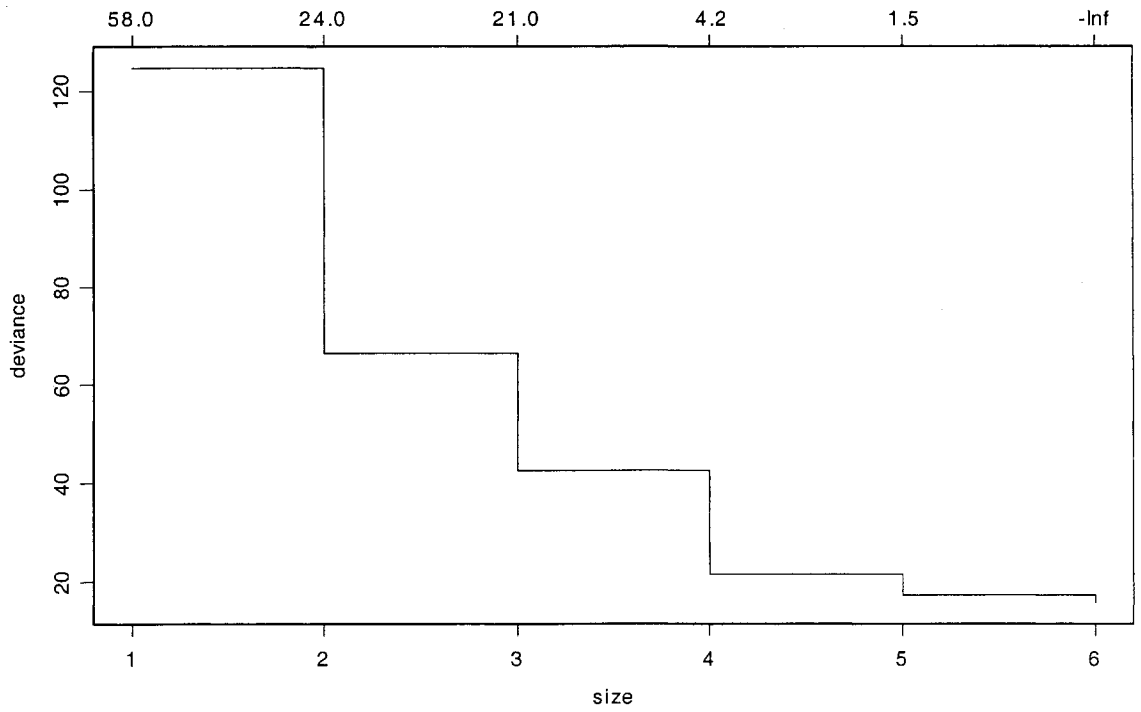
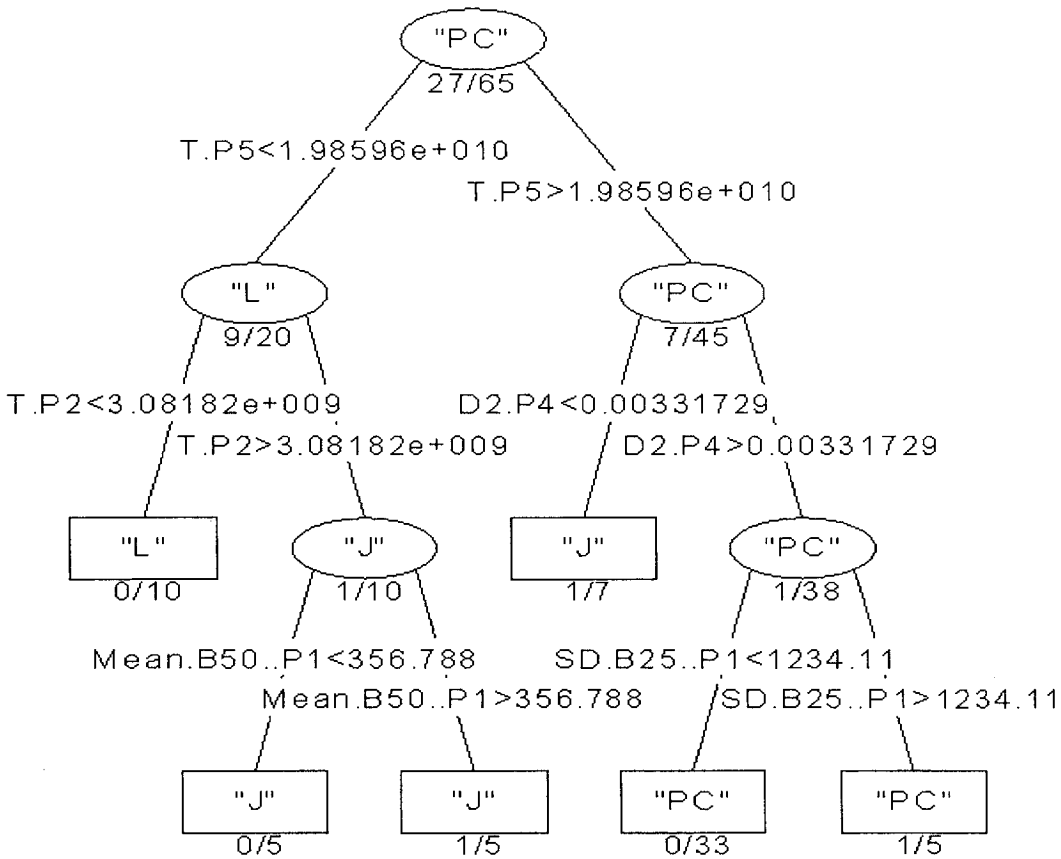


Figure 6-7 Tree method (Pop(English),Country, Jazz, Classic)



PC=Pop/Country(English), J=Jazz, L=Classic

Figure 6-8 Tree method(Pop/Country (English),Jazz, Classic)

True Decision	Pop(English)/Country	Jazz	Classic
Pop(English)/Country	0.84	0.16	0.0
Jazz	0.18	0.82	0.0
Classic	0.0	0.25	0.75

Table 6-7 Confusion table of tree method(Pop(English)/Country, Jazz, Classic)

Discussion:

The results from the overall classification indicated that at a first stage Country and Pop might be better treated together. Table 6-7 is the confusion table produced by the tree classification method.

We find the proportion of recognition of Pop/Country increases to 0.84; the proportion of recognition of Jazz increases to 0.82; the proportion of recognition of Classic is 0.75. It shows our model can classify the three types very well.

There are no misclassification of Pop(English)/Country and Jazz to Classic type. Also no Classic songs are recognized as Pop(English)/Country. The misclassification proportion between Jazz and Pop(English/Country) is lower than 0.2. It is surprising that 0.25 of Classic songs are recognized as Jazz songs.

The confusion table shows that the tree method makes a clear cut between Classic type and the two Pop(English)/Country, and Jazz types; while there are some mixed points between Jazz and Pop(English)/Country.

Our model classifies songs into Pop(English)/Country, Jazz and Classic types, with an overall proportion of 0.803. This is the best result so far.

6.4.3 Adaptive tree approach

In this section, 23 English Pop songs, 14 English Country songs, 16 English Jazz songs and 12 Italian Classic songs are used for classification. Since Classic type has the highest proportion, we distinguish Classic type at the first step. We tried different ways of grouping types at the second step. Firstly, we group Country and Pop together but it leads a poor result. Then, we tried to group Country and Jazz together and it provides a much better result. The adaptive tree is shown in the following diagram.

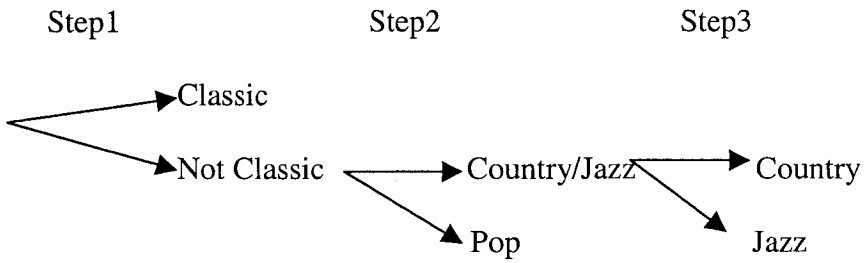


Figure 6-9 Tree method (Pop(English), Country, Jazz, Classic)

Step1:Classify songs as either Classic or not Classic

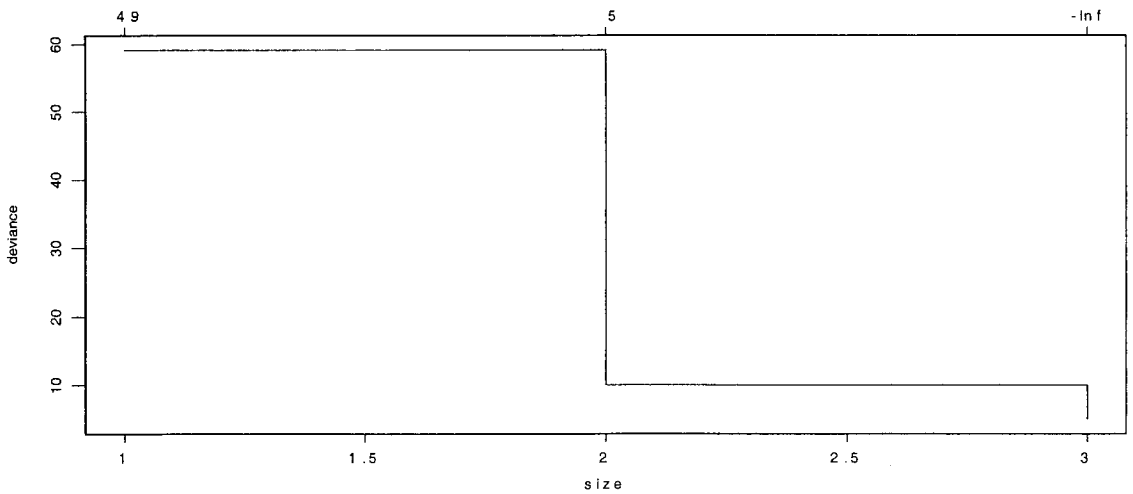
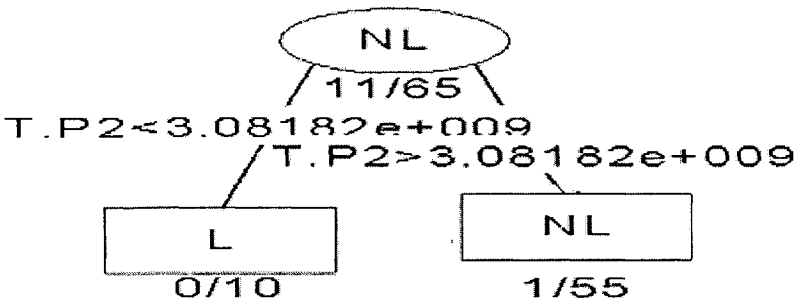


Figure 6-10 Size of Tree(Classic or not Classic)



L=Classic, NL=Not Classic

Figure 6-11 Tree method (Classic or not Classic)

True Decision	Classic	Not Classic
Classic	0.83	0.17
Not Classic	0.0	1.0

Table 6-8 Confusion table in Step 1 of Adaptive tree method

Step2:

Next we distinguish between the two categories Country/Jazz and Pop.

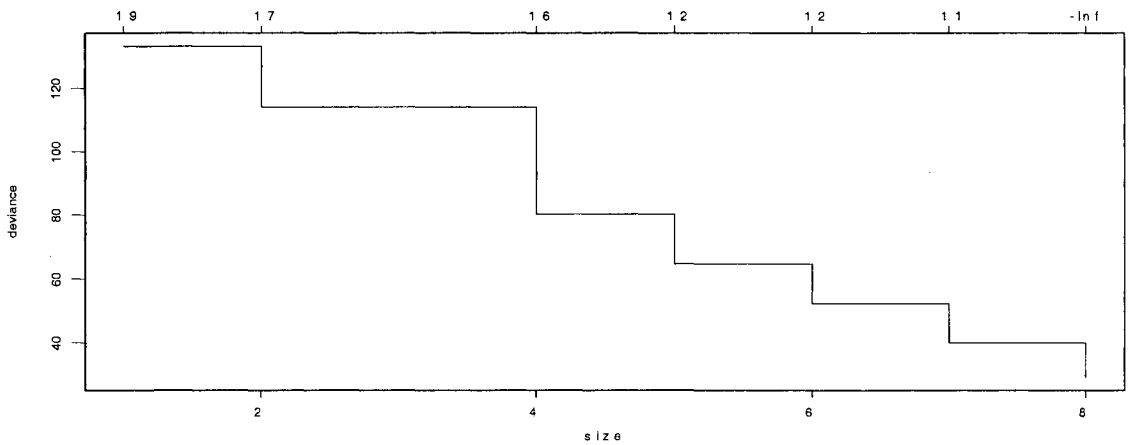
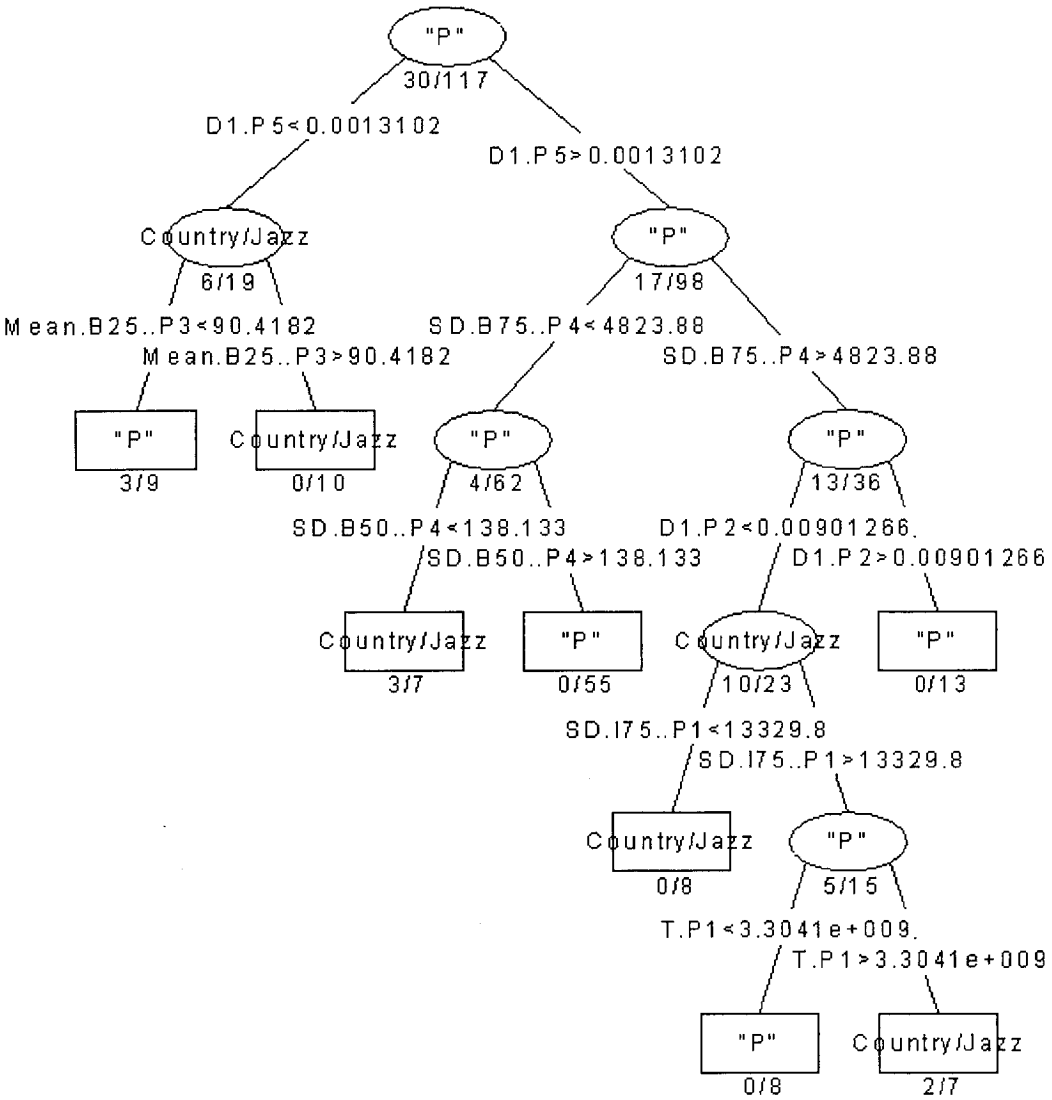


Figure 6-12 Size of Tree (Classic or not Classic)



P=Pop

Figure 6-13 Tree method (Pop and Country/Jazz)

True Decision	Pop	Jazz/Country
Pop	0.82	0.18
Jazz/Country	0.33	0.67

Table 6-9 Confusion table in Step 2 of Adaptive tree method

Step3:

Finally we distinguish between Country and Jazz.

LDA

```
Summary of Classification with Cross-validation

Put into      ....True Group....
Group              C         J
C                  8         1
J                  7        15
Total N           15        16
N Correct          8        15
Proportion     0.533     0.938
N =    31      N Correct =    23      Proportion Correct = 0.742.
```

Table 6-10 LDA in Step 3 of Adaptive tree method

True Decision	Jazz	Country
Jazz	0.92	0.08
Country	0.56	0.44

Table 6-11 Confusion table in Step 3 of Adaptive tree method

A program was written to test this tree approach. (See Appendix 3 Source code 6). The following is the confusion table.

True Decision	Pop	Country	Jazz	Classic
Pop	0.82	0.15	0.03	0.0
Country	0.33	0.30	0.37	0.0
Jazz	0.33	0.04	0.63	0.0
Classic	0.0	0.0	0.17	0.83

Table 6-12 Confusion table of Adaptive tree method

We are also interested in grouping Jazz and Country together to see if we can get a better result.

True Decision	Pop	Country/Jazz	Classic
Pop	0.82	0.18	0.0
Country/Jazz	0.33	0.67	0.0
Classic	0.096	0.074	0.83

Table 6-13 Confusion table of Adaptive tree approach(Pop(English),Country/Jazz,Classic)

Finally, we get a proportion of correct classification equal to 0.773 when Jazz and Country are grouped together.

Discussion:

A cross validation is then performed on the complete adaptive approach for Pop(Engish), Country, Jazz and Classic, this yields a proportion of correct classification equal to 0.64. It shows that the Adaptive tree approach is no better than the tree method.

Here, we are satisfied with the proportion of Pop and Classic. The Country type still only gets a proportion of 0.30. When we try to leave Jazz/Country together, it yields a proportion of correct classification equal to 0.77.

Also, we use the Adaptive tree approach to distinguish songs in the four types, but the Pop songs used are in 4 languages such as English, Chinese, French and Spanish. It still leads to poor result.

6.5 Conclusion

This research provides a series of features to classify voice and signal. It shows a method for feature exploration. Some features are based on the wavelet decomposition while others characterized the time intersection of wave, which reaches one certain energy.

Various statistical methods are used to classify songs on the basis of these features. These include LDA, QDA and the tree method. As well, we use an adaptive tree approach. The tree method provides the best result. It yielded a proportion of correct classification equal to 0.80 with cross validation when Pop and Country are considered as one type and Pop/Country, Jazz songs are in English and Classic songs are in Italian. The adaptive tree approach yielded a proportion of correct classification equal to 0.77 with cross validation under the assumption that Country and Jazz are considered as one type and Pop in English, Country/Jazz in English and Classic in Italian.

All of the approaches of classification show a strong capability in recognizing Pop and Classic music. Comparing with LDA, QDA and the tree method, generally, LDA, QDA are good for pair wise classification while the tree method represents good for overall classification.

By the classification result, it also shows that Country type tends to be influenced by Pop and Jazz nowadays.

We also want to distinguish languages by using those features. However, the classification result is poor and it shows that these features do not present the characteristics of languages.

Chapter 7 Future work

Based on the research we have done, these issues will be considered for further research:

- Can we distinguish different types of music from the same singer?
- Can we distinguish different singers for a given music type?
- Can we characterize the evolution in time of a given singer?
- What about the same singer using different languages?
- What will happen if a song involves several singers?
- Can we recognize the gender of singers?
- What will happen if a song is sung by one person or more than one person?
- What if the approach is based on the whole song instead of 10 seconds clip?
- In future study, verify the consistence of the method as we verify the choice of the clips.
- New features will be defined and explored for classifying languages.

Appendix 1

List of Songs

TYPE	GENDER	LANGUAGE	Singer	Song
POP	FEMALE	ENGLISH	Celine Dion	Misled
POP	FEMALE	ENGLISH	Celine Dion	The Color Of My Love
POP	FEMALE	ENGLISH	Celine Dion	Think Twice
POP	FEMALE	ENGLISH	Celine Dion	The Power Of Love
POP	FEMALE	ENGLISH	Jennifer Lopez	Could This Be Love
POP	FEMALE	ENGLISH	Jennifer Lopez	Should've Never
POP	FEMALE	ENGLISH	Jennifer Lopez	Feeling Good
POP	FEMALE	ENGLISH	Jennifer Lopez	If You Had My Love
POP	FEMALE	ENGLISH	Madonna	La Is la Bonita
POP	FEMALE	ENGLISH	Madonna	Material Girl
POP	FEMALE	ENGLISH	Madonna	Like A Virgin
POP	FEMALE	ENGLISH	Madonna	Lucky Star
POP	MALE	ENGLISH	Elvis	Don't Be Cruel
POP	MALE	ENGLISH	Elvis	Hound Dog
POP	MALE	ENGLISH	Elvis	Love Me Tender
POP	MALE	ENGLISH	Elvis	Heart Break Hotel
POP	MALE	ENGLISH	Enrique Iglesias	Escape
POP	MALE	ENGLISH	Enrique Iglesias	Don't Turn Off The Light
POP	MALE	ENGLISH	Enrique Iglesias	Hero
POP	MALE(GROUP)	ENGLISH	The Beatles	She Loves You
POP	MALE(GROUP)	ENGLISH	The Beatles	If that is what you want
POP	MALE(GROUP)	ENGLISH	The Beatles	I Want To Hold Your Hand
POP	MALE(GROUP)	ENGLISH	The Beatles	Love Me Do
POP	FEMALE	French	Celine Dion	J'irai Ou Tu Iras
POP	FEMALE	French	Celine Dion	Le Ballet
POP	FEMALE	French	Celine Dion	Pour Que Tu M'aimes Encore
POP	FEMALE	French	Celine Dion	Destin
POP	MALE	French	Nino Ferrer	Scopa
POP	MALE	French	Nino Ferrer	Les Cornichons
POP	MALE	French	Nino Ferrer	L'an 2000
POP	MALE	French	Nino Ferrer	La Maison Pres De La Fontaine
POP	MALE	French	Jean-Jacques Goldmans	La Bas
POP	MALE	French	Jean-Jacques Goldmans	Il Changeait La Vie
POP	MALE	French	Jean-Jacques Goldmans	Entre Gris Clair Et Gris Fonce
POP	MALE	French	Jean-Jacques Goldmans	A Quoi Tu Sers
POP	MALE	French	Francis Cabrel	Sarbacane
POP	MALE	French	Francis Cabrel	Rosie
POP	MALE	French	Francis Cabrel	C'est Ecrit
POP	MALE	French	Francis Cabrel	Animal
POP	FEMALE	Chinese	Wang Fei	Ren Jian
POP	FEMALE	Chinese	Wang Fei	When The Moon Get Round
POP	FEMALE	Chinese	Wang Fei	I Am Willing
POP	FEMALE	Chinese	Lin Yi Lian	Have You At Least
POP	FEMALE	Chinese	Xu Mei Jin	Urgent For Love

POP	MALE	French	Francis Cabrel	Animal
POP	FEMALE	Chinese	Wang Fei	Ren Jian
POP	FEMALE	Chinese	Wang Fei	When The Moon Get Round
POP	FEMALE	Chinese	Wang Fei	I Am Willing
POP	FEMALE	Chinese	Lin Yi Lian	Have You At Least
POP	FEMALE	Chinese	Xu Mei Jin	Urgent For Love
POP	FEMALE	Chinese	Li Wen	Cang Long Wo Hu
POP	FEMALE	Chinese	Li Wen	Today To Future
POP	FEMALE	Chinese	Li Wen	Xiang Ni De 365 Tian
POP	FEMALE	Chinese	Zhang Hui Mei	Gei Wo Gan Jue
POP	FEMALE	Chinese	Zhang Hui Mei	Pai Shan Dao Hai
POP	FEMALE	Chinese	Zhang Hui Mei	Hui Gu Niang
POP	FEMALE	Chinese	Zhang Hui Mei	Sister
POP	MALE	Chinese	Zhang Xin Zhe	Nan Yi Kang Ju Ni Rong Yan
POP	MALE	Chinese	Zhang Xin Zhe	Love Is Tide
POP	MALE	Chinese	Zhang Xin Zhe	Tolerance
POP	MALE	Chinese	Zhang Xin Zhe	Ru Guo Ni Yuan Yi
POP	MALE	Chinese	Wu Si Kai	Ai De Gang Qin Shou
POP	MALE	Chinese	Wu Si Kai	Jin Tian Lu Huo Te Bie Len
POP	MALE	Chinese	Wu Si Kai	Beautiful Error
POP	MALE	Chinese	Wu Si Kai	You Are Springwind I Am Rain
POP	FEMALE	SPANISH	Shakira	Antologia
POP	FEMALE	SPANISH	Shakira	Te Necesito
POP	FEMALE	SPANISH	Shakira	Quiero
POP	FEMALE	SPANISH	Shakira	Estoy Aqui
POP	FEMALE	SPANISH	Paulina Rubio	Baila Casanova
POP	FEMALE	SPANISH	Paulina Rubio	Yo No Soy Esa Mujer
POP	FEMALE	SPANISH	Paulina Rubio	Elultimoadios
POP	FEMALE	SPANISH	Paulina Rubio	Yo No Soy Esa Mujer
POP	FEMALE	SPANISH	Thalia	Tu Y Yo
POP	FEMALE	SPANISH	Thalia	Regresa A Mi
POP	FEMALE	SPANISH	Thalia	Quinceanera
POP	FEMALE	SPANISH	Thalia	Vengo De Cana
POP	FEMALE	SPANISH	Thalia	Tuy Yo Cumbia
POP	MALE	SPANISH	Enrique Iglesias	Escapar
POP	MALE	SPANISH	Enrique Iglesias	Hero
POP	MALE	SPANISH	Enrique Iglesias	Tres Palabras
POP	MALE	SPANISH	Enrique Iglesias	No Apaguez La Luz
POP	MALE	SPANISH	Ricky Martin	Bella
POP	MALE	SPANISH	Ricky Martin	Talvez
POP	MALE	SPANISH	Ricky Martin	She Bangs
POP	MALE	SPANISH	Ricky Martin	Jaleo
POP	MALE	SPANISH	Ricky Martin	The Cup of Life
COUNTRY	MALE	English	Alan Jackson	Chattahoochee
COUNTRY	MALE	English	Alan Jackson	Midnight In Mntgomery
COUNTRY	MALE	English	Alan Jackson	She's Got The Rhythm
COUNTRY	FEMALE	English	Anne Murray	Day Dream Believer
COUNTRY	FEMALE	English	Anne Murray	Now and Forever
COUNTRY	FEMALE	English	Anne Murray	Snow Bird

COUNTRY	FEMALE	English	Anne Murray	You Needed Me
COUNTRY	MALE	English	Gordon Lightfoot	Canadian Rail Road Trilogy
COUNTRY	MALE	English	Gordon Lightfoot	Early Morning Rain
COUNTRY	MALE	English	Gordon Lightfoot	For Loving Me
COUNTRY	MALE	English	Gordon Lightfoot	Go Go Round
COUNTRY	FEMALE	English	Shania Twain	Black Eyes Blue Tears
COUNTRY	FEMALE	English	Shania Twain	From This Moment On
COUNTRY	FEMALE	English	Shania Twain	When
COUNTRY	FEMALE	English	Shania Twain	You Are Still The One
JAZZ	MALE	English	Louis Armstrong	Blue Berry Hill
JAZZ	MALE	English	Louis Armstrong	Lazy River
JAZZ	MALE	English	Louis Armstrong	What A Wonderful World
JAZZ	MALE	English	Louis Armstrong	When The Saints Go Marching In
JAZZ	FEMALE/MALE	English	TonyBennett&KDLang	Exactly Like You
JAZZ	FEMALE/MALE	English	TonyBennett&KDLang	I Am Confessing
JAZZ	FEMALE/MALE	English	TonyBennett&KDLang	La Vie En Rose
JAZZ	FEMALE/MALE	English	TonyBennett&KDLang	You Can Depend On Me
JAZZ	MALE	English	Frank Sinatra	It Was A Very Good Year
JAZZ	MALE	English	Frank Sinatra	Some Where In Your Heart
JAZZ	MALE	English	Frank Sinatra	Strangers In The Night
JAZZ	MALE	English	Frank Sinatra	Summer Wind
JAZZ	FEMALE	English	Diana Krall	Devil May Care
JAZZ	FEMALE	English	Diana Krall	Let's Face The Music and Dance
JAZZ	FEMALE	English	Diana Krall	Let's Fall In Love
JAZZ	FEMALE	English	Diana Krall	When I Look In Your Eyes
OPERA	FEMALE	Italian	Kanawa	Ave Maria
OPERA	FEMALE	Italian	Kanawa	Dove Sono
OPERA	FEMALE	Italian	Kanawa	Laudate Dominum
OPERA	FEMALE	Italian	Kanawa	Let The Bright Seraphim
OPERA	FEMALE	Italian	Carreras	T'estimo
OPERA	FEMALE	Italian	Carreras	To Conosco Un Giardino
OPERA	FEMALE	Italian	Carreras	Voce e Note
OPERA	FEMALE	Italian	Domingo	Amor Ti Vieta
OPERA	FEMALE	Italian	Domingo	Memoires De Danton
OPERA	FEMALE	Italian	Domingo	Quiero Desterrar De Tu Pecho El Temor
OPERA	FEMALE	Italian	Pavarotti	Granada
OPERA	FEMALE	Italian	Pavarotti	Nessun Dorma

Appendix 2 Format of Wave file

Offset	Size	Name	Description
0	4	**ChunkID**	Contains the letters RIFF
4	4	**ChunkSize**	36 + SubChunk2Size, or more precisely: 4 + (8 + SubChunk1Size) + (8 + SubChunk2 Size) This is the size of the rest of the chunk following this bytes for the two fields not included in this count: ChunkID and ChunkSize. This is the size of the entire file in bytes minus 8.
8	4	**Format**	Contains the letters WAVE
12	4	**Subchunk1ID**	Contains the letters fmt
16	4	**Subchunk1Size**	16 for PCM.
20	2	**AudioFormat**	PCM = 1 (i.e. Linear quantization) Values other than 1 indicate some form of compression.
22	2	**NumChannels**	Mono = 1, Stereo = 2, etc.
24	4	**SampleRate**	8000, 44100, etc.
28	4	**ByteRate**	SampleRate * NumChannels * BitsPerSample/8
32	2	**BlockAlign**	NumChannels * BitsPerSample/8 The number of bytes for one sample including all channels.
34	2	**BitsPerSample**	8 bits = 8, 16 bits = 16, etc.
36	4	**Subchunk2ID**	Contains the letters data
40	4	**Subchunk2Size**	NumSamples * NumChannels * BitsPerSample/8 This is the number of bytes in the data.
44	*	**Data**	The actual sound data.

Appendix 3
Source code 1 for Tree Classification

```
# There are NA's due to the NA's in the data
as.vector(Clip.pred == unclass(Clip$type))->q
q[is.na(q)] <- FALSE
sum(q)/130
#Classify language
#cross validation
library(MASS)
library(nnet)
p <- vector(length=650)
for(i in 1:650) {
language_Clip$language[-i]
#tree(type~.,data=Clip[-i,c(1,7:21)]) -> Clip.tree
#predict(Clip.tree,Clip[i,c(1,7:21)]) -> q
tree(language~.,data=Clip[-i,c(7:21)]) -> Clip.tree
predict(Clip.tree,Clip[i,c(7:21)]) -> q
p[i] <- which.is.max(q)
}
Clip.pred <- as.factor(p)
plot(Clip.tree)
text(Clip.tree)
rm(p,q)
# There are NA's due to the NA's in the data
as.vector(Clip.pred == unclass(Clip$language))->q
q[is.na(q)] <- FALSE
sum(q)/216
#Classify language in Pop
#cross validation
library(MASS)
library(nnet)
p <- vector(length=435)
for(i in 1:435) {
language_Clip$language[-i]
#tree(type~.,data=Clip[-i,c(1,7:21)]) -> Clip.tree
#predict(Clip.tree,Clip[i,c(1,7:21)]) -> q
tree(language~.,data=Clip[-i,c(7:21)]) -> Pop.Clip.tree
predict(Pop.Clip.tree,Pop.Clip[i,c(7:21)]) -> q
p[i] <- which.is.max(q)
}
Pop.Clip.pred <- as.factor(p)
plot(Pop.Clip.tree)
text(Pop.Clip.tree)
rm(p,q)
```

```
# There are NA's due to the NA's in the data
as.vector(Pop.Clip.pred == unclass(Pop.Clip$language))->q
q[is.na(q)] <- FALSE
sum(q)/435
```

Source code 2 for Doppler algorithm

```
"doppler" <- function(n=256) {
        dop<-vector(mode="numeric",length=n)
        for(i in 1:n) {
                t=i/n
                dop[i]<-sqrt(t*(1-t))*sin((2.1*pi)/(t+0.05))
        }
        return(dop)
}
# data points and levels
nl<-4
n<-1024
# gaussian noise
dop<-doppler(n=n)+rnorm(n)/20
# plot
windows()
plot(dop,type='l')
# dwt
dop.dwt<-dwt(dop,wf="la8",n.levels=nl)
# mra
windows()
dop.mra<-mra(dop,wf="la8",J=nl,method="dwt")
par(mfcol=c(nl+2,1),pty="m",mar=c(5-2,4,4-2,2))
plot.ts(dop,axes=T,ylab="",main="(signal)")
for(i in 1:(nl+1)) { plot.ts(dop.mra[[i]],axes=T,ylab=names(dop.mra)[i]) }
dop.idwt<-idwt(dop.dwt)
#plot.ts(dop.idwt,axes=T,ylab="",main="(idwt)")
sum((dop-dop.idwt)^2)/sum(dop^2)
# cumulative signals plot and diff
windows()
diff<-vector(mode="numeric",length=nl+1)
par(mfcol=c(nl+2,1),pty="m",mar=c(5-2,4,4-2,2))
plot.ts(dop,axes=T,ylab="",main="(signal)")
S<-dop.mra[[nl+1]]
for(i in nl:1) {
        diff[i]=sum((dop-S)^2)/sum(dop^2)
        plot.ts(S,axes=T,ylab=i)
        S<-S+dop.mra[[i]]
}
diff[nl+1]=sum((dop-S)^2)/sum(dop^2)
plot.ts(S,axes=T,ylab="ALL")
```

Source code 3 for Wavelet feature algorithm

```
# load libraries
library(waveslim)
# datafile: filename containing vector of values
# nl: number of levels for the decomposition
# n: chunk size
# wave: wavelet family to be used
# computesum: T=overall ratios, F=matrix, ratios for each chunk
# maxdata: if positive, maximum number of values to read in.
"energy" <-
function(datafile,nl=6,n=1024,wave="la8",computesum=T,maxdata=0,col=1) {

        df <- file(datafile,"r")
        d<-scan(df,n=2*n,quiet=T)
        data <- d[2*(0:(n-1))+col]
        rm(d)
        l<-length(data)
        i<-1
        while(l==n) {
                # dwt
                data.mra<-mra(data,wf=wave,J=nl,method="dwt")
                # v will hold the energy values
                v<-vector(length=nl+2)
                v[nl+2]<-0
                for(j in 1:(nl+1)) {
                        v[j]<-sum(data.mra[[j]]^2)
                }
                # divide by sum of energies
                s<-sum(v)
                if(s>0) {
                        for(j in 1:(nl+1)) {
                                v[j]<-v[j]/s
                        }
                }
                v[nl+2]<-s
                rm(s)
                # add to matrix m
                if(i==1) { m <- t(v) }
                else { m<-rbind(m,v) }
                rm(v)
                # read in next block
                d<-scan(df,n=2*n,quiet=T)
                data <- d[2*(0:(n-1))+col]
                rm(d)
                l<-sum(!(is.na(data)))
```

```
        i<-(i+1)
        if(maxdata>0 & (i*n)>maxdata) { l<-0 }
    }
    l<-i-1
    # take mean of energies and proportions
    m<-rbind(m,apply(m,2,mean))
    # compute ratios - obsolete (overall computation)
    # for(i in 1:nl) { m[,i] <- (m[,i]/m[,nl+1]) }
    # names
    m<-as.data.frame(m)
    row.names(m)<-c(1:(l),"all")
    names(m) <- c(paste("D",1:(nl),sep=""),paste("S",nl,sep=""),"Total")
    close(df)
    rm(l,i,data)
    if (computesum==T) { return(m[nrow(m),1:ncol(m)]) }
    else { return(m[1:(nrow(m)-1),1:ncol(m)]) }
}
```

Source code 4 for time based feature 1 algorithm

```
"TD" <- function (FileName, ratio)
{
#load file to matrix
print("start  program")
#       datafile<-
"c://ChengWu//Data//Database//ClassicOpera.F.IT.Kanawa.AveMaria.Piece1.txt"
        datafile<-FileName
        df <- file(datafile,"r")
        d<-scan(df,,quiet=T)
        n<-length(d)/2
        data <- d[2*(0:(n-1))+1]
        len<-length(data)
print("initial data")
        overflag<-0
        wavenum<-0
        i<-1
        q<-max(data)*ratio
        while(i<len+1)
        {
                if(data[i]>q)
                        overflag<-1
                else if(overflag==1)
                        {
                                overflag<-0
                                wavenum <-wavenum+1
          print(wavenum)
                        }
                i<-i+1
        }
print("Create vector contains distances between wavers which are over 3/4 maxium")
pos<- seq(1,wavenum,1)
i <- 1
while(i< wavenum+1)
{
        pos[i]<-0
        i<-i+1
}
print("wavenum")
print(wavenum)
print("Get position where wave is higer than 3/4 maxium")
i<-1
j<-1
while(i<len+1)
{
```

```
        if(data[i]>q)
        {
                if(overflag==0)
                {
                        overflag<-1
                        pos[j]=i
                        j<-j+1
        print("wave position")
                        print(i)
                }
        }
        else
{
 if(overflag==1)
                {
                        overflag<-0
                }
        }
        i<-i+1
}
print("Caculate distance between two waves where they are higher than 3/4maxium")
i<-1
dis<- seq(1,wavenum,1)
i <- 1
while(i< wavenum+1)
{
        dis[i]<-0
        i<-i+1
}
i<-1
while(i<wavenum)
{
  a=pos[i+1]-pos[i]
  dis[i]=a
 print(dis[i])
i<-i+1
}
disMean=mean(dis)
disSD=sd(dis)
Return=cbind(disMean,disSD)
close(df)
Return
}
```

Source code 5 for time based feature 2 algorithm

```
"TDfigure2" <- function (FileName, ratio)
{
#load file to matrix
print("start  program")
#       datafile<-"c://ChengWu//Data//Database//a.txt"
        datafile<-FileName
        df <- file(datafile,"r")
        d<-scan(df,,quiet=T)
        n<-length(d)/2
        data <- d[2*(0:(n-1))+1]
        len<-length(data)
        print("initial data")
        overflag<-0
        i<-1
        q<-max(data)*ratio
        pos<-0
        while(i<len+1)
        {
                        if(data[i]>q)
                        {
                                if(overflag==0)
                                {
                                        pos=cbind(pos, i)
                                        print("up position")
                                        print(i)
                                        overflag<-1
                                        #wavenum <-wavenum+1
                        }
                        }
                        else
                        {
                                if(overflag==1)
                                {
                                        pos=cbind(pos, i)
                                        print("down position")
                                        print(i)
                                        overflag<-0
                                }
                        }
                        i<-i+1
        }
print("Caculate distance that wave remains higher than 3/4maxium")
i<-1
j<-1
```

```
dis<-NULL
while(i< length(pos))
{
  a=pos[i+1]-pos[i]
  dis[j]=a
 print(dis[j])
i<-i+2
j<-j+1
}
disMean=mean(dis)
disSD=sd(dis)
result=cbind(disMean,disSD)
close(df)
result
}
```

Source code 6 for Verify Adaptive tree approach algorithm

```
#Verify Step1: Classic, NoneClassic Step2: NoneClassic-> Pop,NonePop Step3: NonePop-
> Jazz, Country
        library(MASS)
        library(nnet)
        p <- vector(length = nrow(Clip))
        len_nrow(Clip)
        temp <- Clip$Type
        tmp_Clip$type1
        for(i in 1:nrow(Clip)) {
                type <- tmp[ - i]
                Clip.tree <- tree(type ~ ., data = Clip[ - i, c(10:84)])
                q <- predict(Clip.tree, Clip[i, c(10:84)])
                p[i] <- which.is.max(q)
        }
# get the number of classic type songs
        orgclass<-sum(unclass(tmp)==1)
#get the number that can't predict classic songs
        predclass<-sum(p==1)
        temp2<-(unclass(tmp)==1)-(p==1)
        corrclass_orgclass-sum(temp2==1)
        Clip$p<-p
        Clip1_Clip[Clip$p!=1,]
        tmp_Clip1$type2
        p <- vector(length = nrow(Clip1))
        for(i in 1:nrow(Clip1)) {
                type <- tmp[ - i]
                Clip.tree <- tree(type ~ ., data = Clip1[ - i, c(10:84)])
                q <- predict(Clip.tree, Clip1[i, c(10:84)])
                p[i] <- which.is.max(q)
        }
        orgpop_sum(unclass(tmp)==2)
        predpop_sum(p==2)
        temp2_(unclass(tmp)==2)+(p==2)
        corrpop_sum(temp2==2)
        Clip1$p_p
        Clip2_Clip1[Clip1$p!=2,]
        tmp_Clip2$type3
        p <- vector(length = nrow(Clip2))
        for(i in 1:nrow(Clip2)) {
                type <- tmp[ - i]
                Clip.tree <- tree(type ~ ., data = Clip2[ - i, c(10:84)])
                q <- predict(Clip.tree, Clip2[i, c(10:84)])
                p[i] <- which.is.max(q)
```

orgcountry_sum(unclass(tmp)==1)
predcountry_sum(p==1)
temp2_(unclass(tmp)==1)-(p==1)
corrcountry_orgcountry-sum(temp2==1)
orgjazz_sum(unclass(tmp)==2)
predjazz_len-res2-res5-res8
temp3_p+unclass(tmp)
corrjazz_sum(temp3==4)
res_c(orgclass,predclass,corrclass,orgpop,predpop,corrpop,orgcountry,predcountry,corrcountry,orgjazz,predjazz,corrjazz)
resratio_c(corrclass/orgclass,corrpop/orgpop,corrcountry/orgcountry,corrjazz/orgjazz)

Appendix 4 PCM Code

	Voltage	Multiple of D	PCM Code
1	-0.08 v	-0D	0000
2	-0.275 v	-1D	0001
3	-0.55 v	-2D	0010
4	-0.825 v	-3D	0011
5	- 1.1 v	-4D	0100
6	- 1.375v	-5D	0101
7	-1.65 v	-6D	0110
8	1.925 v	-7D	0111
9	0.02 v	0D	1000
10	0.275 v	1D	1001
11	0.55 v	2D	1010
12	0.825 v	3D	1011
13	1.1 v	4D	1100
14	1.375v	5D	1101
15	1.65 v	6D	1110
0	1.925 v	7D	1111

Table Transformation of voltage to PCM code

References

[1] Rabiner, L.R. and Schafer, R.W., *Digital Processing of Speech Signals*, Prentice-Hall, 1978.

[2] Venables W.N., Smith D.M. and the R Development Core Team, *An Introduction to R*, version 1.5.0, 2002.

[3] Bruce, A. and Gao, H.-Y., *Applied Wavelet Analysis with S_PLUS,* Springer, 1996.

[4] Donald F. Morrison, *Multivariate Statistical Methods Second Edition ,1976*

[5] R Development Core Team, *What is R?,* http://www.r-project.org/,Current Version: 2.0.1 (November 2004)

[6] Publisher: *8to32.com* , http://www.8to32.com/, December 14, 2002

[7] Mark D.S. and John G.H.,Computational Neuro-Engineering Laboratory, University of Florida, *Increased MFCC Filter Bandwidth for Noise-robust Phoneme Recognition*

[8] Hastie T, Tibshirani R and Friedman J *The Elements of Statistical Learning Data Mining, inference, and Prediction,* 2001

[9] R Development Core Team, *R Manual,* http://www.r-project.org/,Current Version: 2.0.1 (November 2004)