

*Annotation Based Classification of Audio
Data: Comparing Naïve Bayesian & Decision Tree Classifiers*

*Dissertation submitted to the Jawaharlal Nehru University in partial fulfillment
of the requirement for the award of the degree of*

MASTER OF TECHNOLOGY
IN
COMPUTER SCIENCE AND TECHNOLOGY

By

Arun Kumar Gautam



SCHOOL OF COMPUTER AND SYSTEM SCIENCES
JAWAHARLAL NEHRU UNIVERSITY
NEW DELHI-110067, INDIA

JULY 2007

DECLARATION

This is to certify that the dissertation titled “**Annotation Based Classification of Audio data: Comparing Naïve Bayesian & Decision Tree Classifiers**”, which is being submitted to the **School of Computer & Systems Sciences, Jawaharlal Nehru University, New Delhi**, in partial fulfillment of the requirements for the award of **Master of Technology in Computer Science & Technology** is a bonafide work carried out by me.

The contents of the dissertation have not been submitted for the award of any other degree or diploma.

Arun Kumar Gautam

ARUN KUMAR GAUTAM
M.Tech, SC & SS, JNU,
New Delhi - 110067.

CERTIFICATE

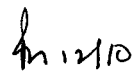
This is to certify that this dissertation entitled “**Annotation Based Classification of Audio data: Comparing Naïve Bayesian & Decision Tree Classifiers**” submitted by **Mr. Arun Kumar Gautam**, to the School of Computer and Systems Sciences, Jawaharlal Nehru University, New Delhi, for the award of degree of **MASTER OF TECHNOLOGY**, is a bonafide work carried out under my supervision.

This research work is original and has not been submitted, in part or in full, to any other University or Institution for the award of any other degree.



(Dr. Sonajharia Minz)

Supervisor
Associate Professor
School of Computer & System Sciences
JNU, New Delhi-67



(Prof. Parimala N.)

Dean
School of Computer & System Sciences
JNU, New Delhi-67

Prof Parimala N:
Dean
School of Computer & Systems Sciences
JAWAHARLAL NEHRU UNIVERSITY
NEW DELHI-110067

Acknowledgments

I owe a great debt of gratitude to many people who have helped make the research work in this dissertation possible.

First and foremost, I thank my supervisor, Dr. Sonajharia Minz, for her guidance, support, and encouragement throughout year. I am also grateful for her wisdom, understanding, and suggestions throughout the course of this study.

I owe my heartfelt gratitude to Prof. S. Balasundaram, former Dean, Prof. Parimala N., Dean, School of Computer & Systems Sciences, Jawaharlal Nehru University, New Delhi, for his kind and active cooperation during the course of study.

I would like to thank to entire faculty and staff of SC&SS for their cooperation during the course of study.

I am thankful to my parents, without whose blessing and support it would not possible to complete this work.

I also extend my thanks to all of my classmate and my senior especially Dr. Girish & Mr. Ahmad for their warmth, care and morale support.

Arun Kumar Gautam
(Arun Kumar Gautam)

Dedicated to

My Parents.....

Contents

Declaration	i
Certificate.....	ii
Acknowledgement.....	iii
1. Introduction	
1.1 Introduction to Audio Data.....	1
1.1.1 Analog Audio Data.....	1
1.1.2 Digital Audio Data.....	1
1.2 Storage of Audio data.....	3
1.3 Audio Classification.....	3
1.4 Problem Description.....	4
1.5 Organization of Report.....	5
2. Literature Review	
2.1 Audio Data Analysis.....	6
2.2 Application of Audio Data.....	8
2.2.1 Speech Recognition.....	8
2.2.2 Audio Analysis for Surveillance Applications.....	9
2.2.3 Birds Sound Detection.....	9
2.3 Multimedia data mining.....	10
3. Basic Concepts	
3.1 Audio Segmentation.....	17
3.1.1 Detecting Segmentation Boundaries.....	17
3.1.2 Classification of Segments.....	17
3.1.2.1 Detecting Silence.....	17
3.1.2.2 Separating Sounds with Music Component.....	18
3.1.2.3 Detecting Harmonic Environmental Sounds.....	19
3.1.2.4 Distinguishing Pure Music.....	19
3.1.2.5 Distinguishing Song.....	20
3.1.2.6 Separating Speech with Music Background and Environmental sound with Music Background.....	20

3.1.2.7 Distinguishing Pure Speech.....	20
3.1.2.8 Classifying Non-harmonic Environmental Sounds.....	21
3.2 Detection of Key Audio Element.....	22
3.3 Audio Features.....	22
3.3.1 Zero-Crossing Rate (ZCR).....	23
3.3.2 Short-time Energy.....	23
3.3.3 Root Mean Square.....	24
3.3.4 High Feature Value Ratio.....	24
3.3.5 Low Feature Value ratio.....	25
3.3.6 Spectrum Centroid.....	26
3.3.7 Spectrum Spread.....	27
3.3.8 Spectral Flux.....	27
3.3.9 Spectral Rolloff Frequency.....	27
3.3.10 Mel Frequency Cepstral Coefficient (MFCC).....	28
3.4 Annotation.....	28
3.5 Classification Algorithm/Technique.....	28
3.5.1 Gaussian Classifier.....	28
3.5.2 K-Nearest Neighbor Classifier.....	29
3.5.3 C4.5 Classifier.....	30
3.5.4 Bayesian Classifier.....	31
3.5.5 Hidden Markov Model (HMM).....	32
3.5.6 Multi Layer Perception (MLP) Neural Network.....	32
4. Proposed Method	
4.1 Framework for Audio Classification.....	33
4.1.1 Audio Data Representation.....	35
4.1.2 Classification.....	35
4.2 Performance Evaluation.....	38
5 Experiments & Results	
5.1 Experimental Objective	41
5.2 System Characteristics.....	41

5.3 Databases.....	42
5.4 Design of Experiments.....	42
5.5 Results and Analysis.....	43
6 Conclusion	47
References	48

Chapter 1

Introduction

1.1 Introduction to Audio Data

Personal and public collections of digital music have become increasingly common over the recent years. The amount of digital music available on the Internet has increased rapidly at the same time. As the amount of data increases, efficient management of the digital content is becoming more and more important. However, most of the indexing and labeling of the music is currently performed manually, which is time-consuming and expensive [20].

Training the computers to recognize sounds has been a popular research task since computers became useful tools for analyzing data. Audio analysis research is inter and multi-disciplinary, and different parts of the sound classification task are interesting for different research domains, including physics, psychology, audiology, music, cognitive science, and philosophy. “If a tree falls in the forest, and there is no-one around to hear it, does it make a sound?” If sound does not exist apart from our perception of it, then the understanding of the physical properties of sound is intimately connected to the understanding of our perception of sound, and any study of sound must also include a study of the perceptual science of sound as well [3].

Audio data is broadly stored into the two formats:

1. Analog Audio Data
2. Digital Audio Data

1.1.1 Analog Audio Data

Analog audio data format is the oscillating voltages that are used to represent the original sounds. Generally they are expressed in the waveform. Analog audio data are the original sound signal. Analog audio data are represented in the continuous waveform. Analog audio signal is a variable signal continuous in both time and amplitude. The representation of analog audio data is represented in figure 1.1.

1.1.2 Digital Audio Data

Digital audio data is the representation of audio data into digital formats. Analog audio data are converted into the digital format by using the process of sampling and quantization, where

sampling is defined as reducing a continuous signal to a discrete signal and quantization is the

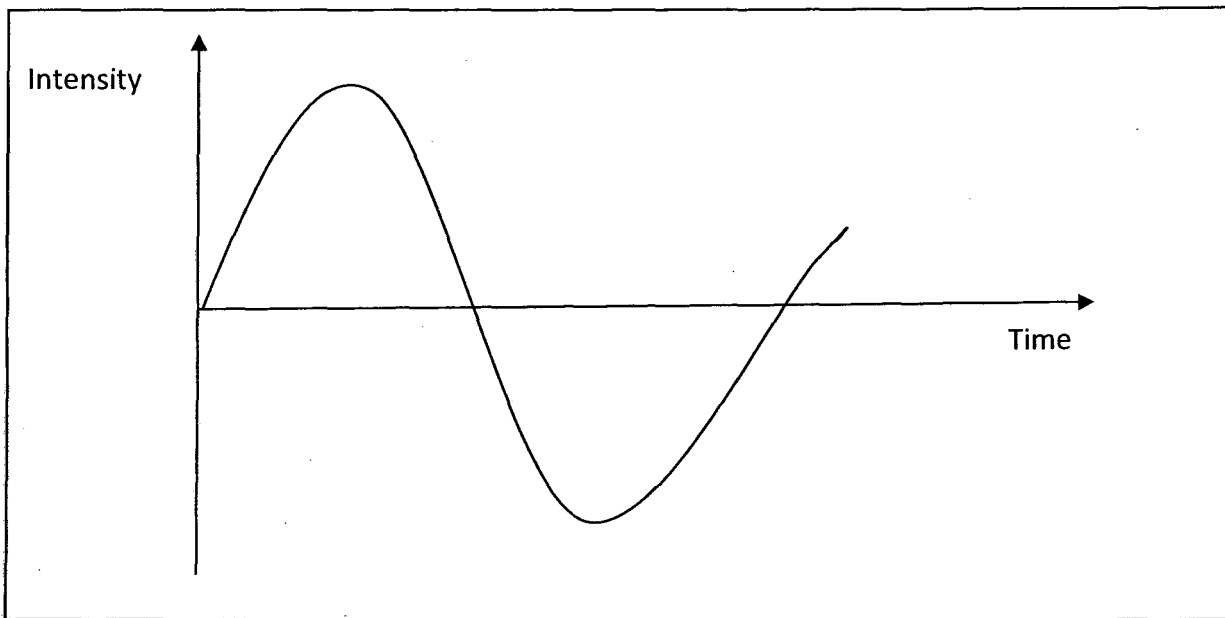


Figure 1.1: Analog Audio Signal

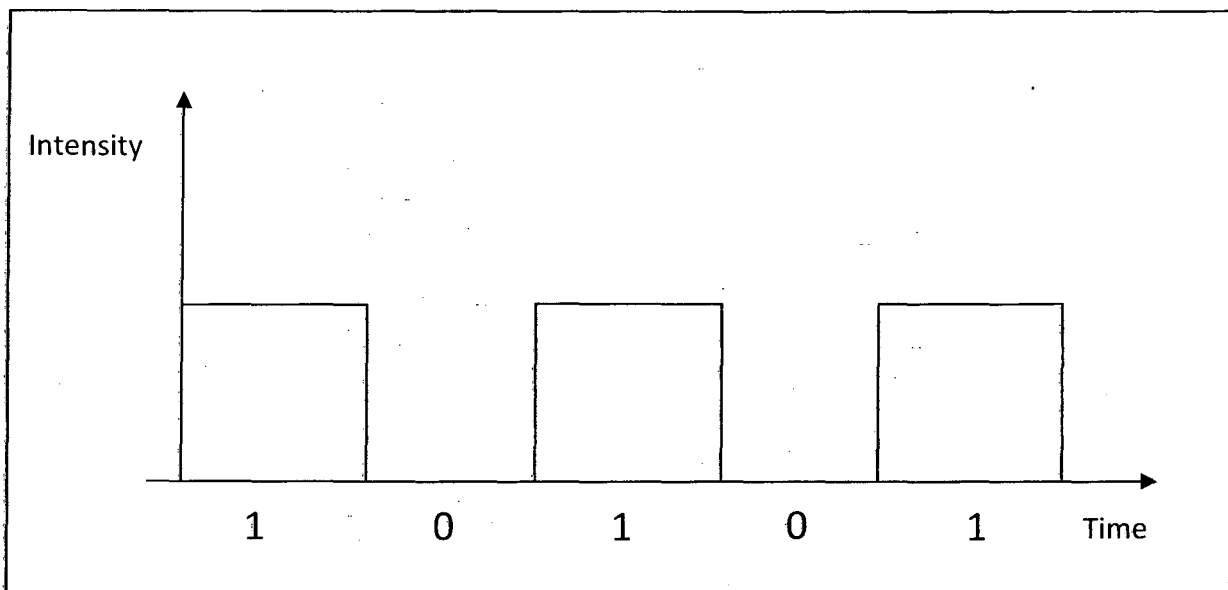


Figure 1.2: Digital Audio Signal

process of approximating a continuous range of values into relatively small set of discrete symbols integer values. The representation of analog audio data is shown in figure 1.2.

1.2 Storage of Audio Data

In [5], Storage of audio data is achieved by using audio file format. Audio file format is defined as, a container format for storing audio data in the digital format that are read by the computer where analog audio data are captured by the microphone, telephone. Container format are the container for computer file format used to store various types of data. Generally a file format is a structure that contains the information about the data and data itself. Audio library supports some types of audio data formats that are Microsoft RIFF (Resource Interchange File Format)/waveform, which is chunk based and International Consultative Committee for Telephone and Telegraph (CCITT). They use filename extension to represent the audio data format. These filename extensions are the suffix to the name of the audio data separated by a dot (.). Some of the filename extensions are:

- WAV – RIFF file format used on windows platform,
- AIFF (Audio Interchange File Format) - IFF format used for MAC OS platform. IFF is
 - the first platform independent container audio format.
- AVI (Audio Video Interleave) – the standard Microsoft windows container based on RIFF,
- u- Mu law,
- al – A Law, etc.

1.3 Audio Classification

In audio classification, classification of audio data has been done by the extracted feature into the different classes. The basic classification is done when there is unknown input data are assigned to a class. Such assignments are done on the basis of or by applying decision rule.

Generally, classification algorithm are divided into two types, one is supervised algorithm while the second one is the unsupervised algorithm. In a supervised algorithm, a labeled set of training sample is used to train the algorithm whereas in case of unsupervised classification algorithm the data is grouped into some clusters without the use of labeled training samples [1].

Parametric and non-parametric classification is another way of categorizing classification algorithm. The functional form of the feature vectors of each class is known in parametric method. On the other hand, in non-parametric method no specific functional form is assumed in advance, instead the probability density is rather approximated locally based on training samples or training data.

In the literature application of some classification algorithms and techniques have been evident. In [1], K-Nearest Neighbor classification is performed on the training as well as test data. They are used for the two-class classification problem in this paper. In [1], General Mixture model classifier is also used for the same problem and the comparison is done on the basis of different feature. In [9], the classification of speech and music has been done by using the Multi-Layer Perception (MLP) Neural network that uses the mean and variance of the discrete wavelet transform. In [18], they have used the three model classifier K-NN, Gaussian Mixture Model, and Hidden Markov Model for the classification of audio data.

Generally people talk about the discriminating ability or the recognition of the different types of voices. One may recognize a particular sound distinctly but in the presence of mixture of sound it becomes a tough task. Mixture of sound means the combinations of speech, music, noise and etc. Recognizing a sound and to be able to determine if it is sound of a particular instrument etc. is mainly the task of classification of audio data.

1.4 Problem Description

Audio data are generally in mixed format such as the speech, music, environmental sounds, noise etc. The task of distinguishing between different categories of audio data is very difficult. For example, in a party a number of sounds may be audible such as environmental sound, instrumental sound, and vocal. The classification of audio data may be of use in video and multimedia application. In data such as audio and image data, data representation and feature extraction are the major data preparation steps. In data representation annotation is used as the representation of metadata and contents (extracting element by segmentation) of audio data. Annotation is based on the features (low level) extracted from the segments of the audio stream. For a given audio stream data each of the features could in turn be described by its characteristic vector. Therefore it may be possible to analyze an audio stream after performing its annotation

with a computed value for each of these characteristics. Although there are a wide range of features that may be extracted from an audio data file, in the present work the time based feature are identified for modeling a classifier. In this dissertation the problem of modeling a classifier using the database of features extracted from low level features of audio data collection, is undertaken. Considering that audio data may require some data preparation, audio data representation is considered as a sub-problem before taking up the problem of modeling the classifier. Further two popular techniques namely Naïve Bayesian and Decision Tree based classification models are investigated to identify a better suited one.

The objectives of the work are to identify annotation based method for audio data representation or feature extraction of audio data. The comparison between them prefers the best classifiers for the audio data of the two algorithms, Naïve Bayesian and C4.5 for decision tree induction, identify the better suited algorithm for classification of audio data based on their performance.

1.5 Organization of the Report

Chapter 2 describes about the literature survey of the audio data, containing audio data analysis and various types of application of audio classification. Chapter 3 describes about the basic concepts related to the audio data, for example segmentation, annotation, and audio classification. Chapter 4 describes about the proposed work we are doing in this dissertation that is the framework for classification and performance evaluation with different parameter. Chapter5 describes about the experimental work we have performed regarding the audio classification of audio data and the result we have obtained based on the experiments. Chapter 6 describes the conclusion.

Chapter 2

Literature Review

2.1 Audio Data Analysis: A Review

The term audio is treated as a sound that is defined as the disturbance of mechanical energy that propagates through the matter as a longitudinal wave. The characteristics of sound are frequency, amplitude, wavelength, period and speed. Sounds are used to hear by the humans and animals by their ear. Generally human are able to hear the sound of frequency range of 20 Hz to 20 KHz. Sound generates from the different sources. These sources are the music instruments, and the voices of human and animals and some of the natural resources. Some term related to audio data are [25]:

- **Sampling Rate**

Sampling rate is the number of digital samples used to represents one second of analog signal. Accuracy of the audio signal reproduction depends upon the number of samples. Greater number of the samples shows the more accurate audio signal reproduced. Sampling rate varies in different types of audio data. For example a sampling rate of 8 kHz can produce a human voice with adequate clarity but it does a poor job on the music. It needs up to 44.1 kHz of sampling rate to represent the music. The music is recorded at this sampling rate. The audio track needed 48 kHz of sampling rate for recording. The sampling rate of audio data ranges from 5.5 kHz to 48 kHz.

- **Quantization**

It is used to convert an analog audio signal into digital audio signal. First of all get the samples of the analog signal by using the sampling theory and divide the whole signal into a finite set of discrete time interval according to the samples.

- **Multiple Channels**

There are more than one channel exists in the audio files. These may be of single channel (mono) or two (stereo) channel audio files. Two channels in a stereo are typically are interleaved on a sample by sample basis.

• **Byte Ordering**

One problem with supporting files across a heterogeneous environment is that the byte ordering of the native hardware may be different. The Audio Library can determine the byte ordering of the audio hardware connected to the system. However, there is no easy way to determine the byte ordering of the audio data in a file that was imported from elsewhere. Therefore, it is forced to make assumptions. The Audio Library assumes that all RIFF/Waveform files use least-significant-byte-first order and that all other files use most-significant-byte-first. This applies even to the files created by our audio tools.

Audio data are the waveform and numerical representation of the sound that means in the analog and digital formats. Audio data are classified in the categories like speech, music, silence and noise. Different compression techniques are used to represent the audio data. These are the lossy compression, loss-less compression, μ -law compression, and A-law compression.

i. Lossy audio format

Lossy compression typically achieves far greater compression than lossless audio compression technique. Lossy file format are based on psychoacoustic model that remove the audio data that cannot or hardly hear by the human e.g. a low volume sound after a high volume sound. Psychoacoustic model analyses the audio signal and computes the amount of noise masking available as a function of frequency. While removing that type of non-hearable sound or we can say that reducing the number of bits used to code a signal increases the amount of noise in that signal. These noises are hiding by using very small numbers of bits to code the high frequencies of most signals. MP3 and Vorbis are the popular example. The target rates of data are achieved in the lossy audio compression by bit rate. They are used in the application such as digitally compressed data are used in most videos DVDs; digital television; streaming media on the internet; satellite and cable radio and increasingly in terrestrial radio broadcasts [25].

ii. Loss-less compression

Lossless audio format provide a compression ratio of 2:1, sometimes more. In exchange of their lower codec they don't destroy any original data. This means when audio data is uncompressed for playing, the sound produced will be identical to those of the original sample. These are

generally used for keeping audio data in permanent collection. Lossless formats such that Dolby TrueHD is also being used for high definition DVD format. Lossless audio codec have no quality issues, so the usability can be estimated by four things [25]. They are:

- a. Speed of compression and decompression
- b. Degree of compression
- c. Software and hardware support
- d. Robustness and error correction

iii. A-Law Compression and μ -Law Compression

Human ear senses the sound based on the logarithmic function. It only loses that information which the ear would not hear anyway, and gives good quality result for the both speech and music. It needs less processing power for the compression because usually the compression ratio is not very high in this case. It is the international standard telephony encoding format that is also known as ITU standard. It is commonly used in North America and Japan for ISDN 8 kHz sample rate, voiced grade, digital telephone service [25].

2.2 Application of Audio Classification

2.2.1 Speech recognition

In [22], Speech recognition is the process of converting an acoustic signal, captured by a microphone or telephone, into a set of words. The recognized words can be the final results, as for the application such as command and control, data entry, and document preparation.

In these components, at the level of signal representation, it highlight perceptually important speaker-independent feature of the signal and de-emphasized speaker-dependent characteristics. Acoustic model, lexical model and language model are used to handle the phoneme in different context of the linguistic environment. Generally Hidden Markov Model is used to classify the phonemes and finally the recognized word is searched through this whole architecture, which is shown in figure 2.1.

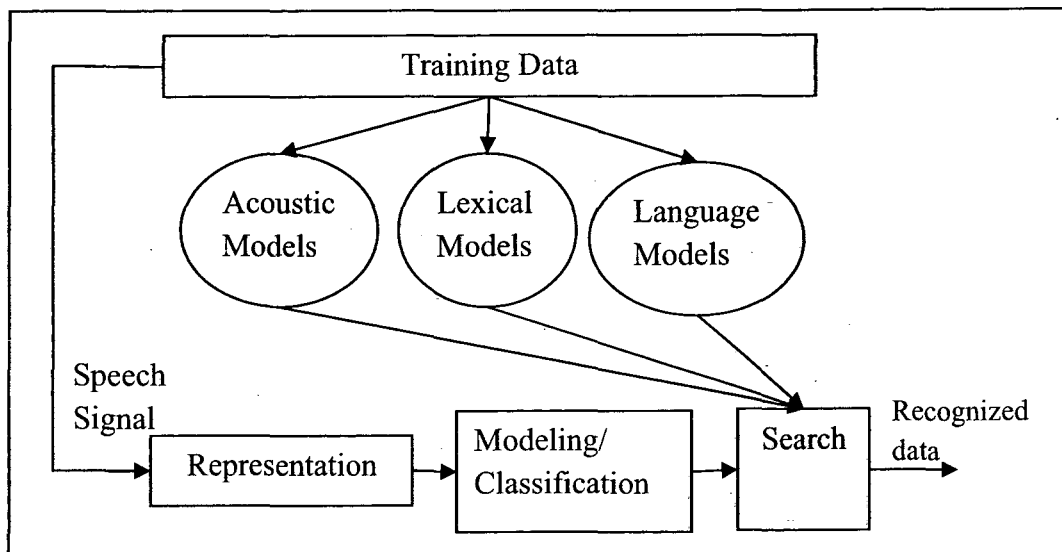


Figure 2.1: Component of Typical Speech Recognition System [22]

2.2.2 Audio Analysis for Surveillance Applications

According to this a hybrid solution is proposed for the detection of crime in the elevator which consists of two parts. One is to perform unsupervised audio analysis and another part is to perform the analysis using an audio classification framework obtained from off-line analysis and training. This system is capable of detecting new kinds of suspicious audio events that occur as outlier against a background of usual activity. It uses the Gaussian Mixture Model to model the background sounds and updates the model incrementally as new audio data arrives. New types of suspicious events can be detected as deviants from this usual background model [13].

2.2.3 Birds Sound Detection

Birds have evolved a vocal apparatus, called the Syrinx, which is not located in the larynx at the end of trachea like in mammals, but much closer to the lung at the divergence of the trachea into the primary bronchi. When the air coming from the lungs passes through it, the membranes of the syringes vibrate and generate sounds. The variation in pitch and frequency are controlled by syringeal muscles a bird has, the more complex the songs he can produce. The structure of the syrinx varies across the species, and as such branches can produce independently a sound, they can produce a far greater variety of sounds than human can do.

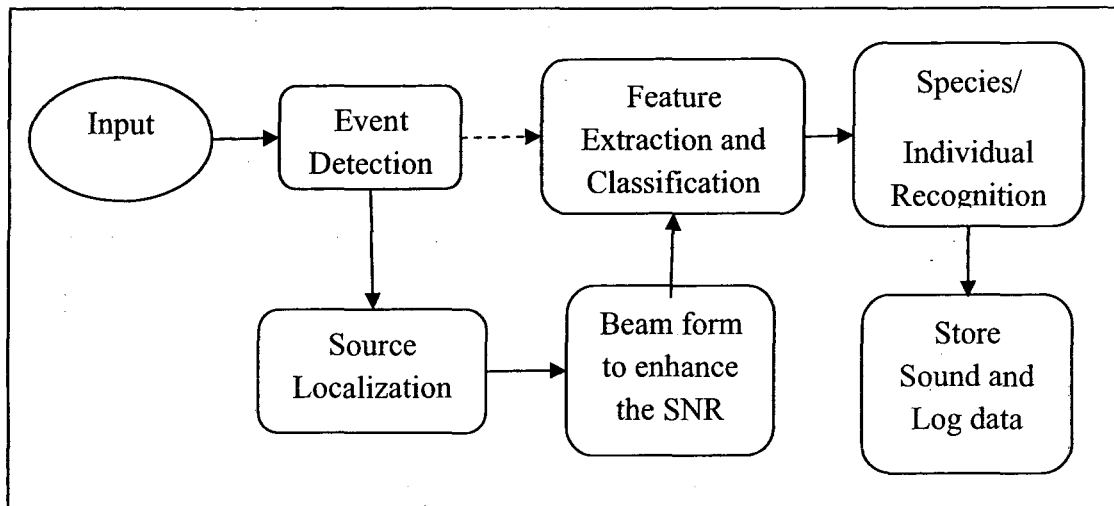


Figure 2.2: Framework for Bird Sound Detection [21]

First process is to for the data acquisition, which is used to record audio data. There are four methods to record data; first is the classic field recordings on DATs using directional microphone. Second is the live internet streaming based on Omni directional microphones. Third one is the microphone array connected to a laptop where data is recorded. And fourth is the microphone array connected to a sensor node to process data in real-time. Second process is to detect sound from the input source given by the first process. Song detection has been done by using Constant False Alarm Rate Detection algorithm. According to that algorithm, it allows identifying high energy segments in continuous stream of audio data and also used to separate out between bird songs and background noise. Third process is the source localization and beam forming, that is the method to localize the bird singing a segment detected by a second process, and improve the song quality recorded by using beam forming. Final process is the classification of bird songs, that classify the species and/or individual that produced each sound file created by third process and finally stores the data [21].

2.3 Multimedia data mining

Most of the content of this topic has been obtained from [2]. Multimedia data mining consist of the two things: first one is the multimedia database management system and second thing is the multimedia database.

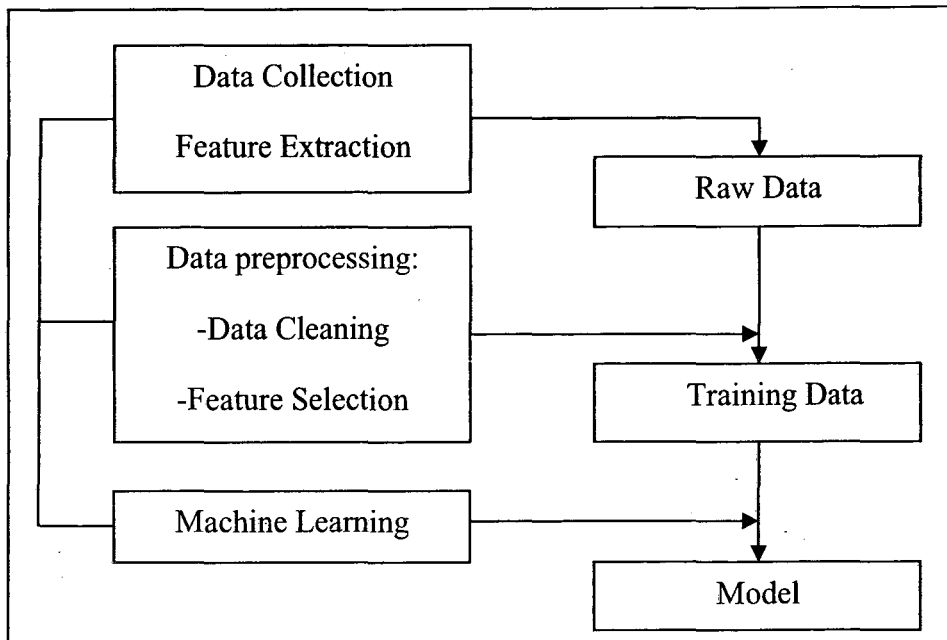


Figure 2.3: Multimedia Mining Process [2]

Multimedia database is generally managed by multimedia database management system, is the collection of multimedia data. Recently the major researches in multimedia data mining are growing in the field of text, image, audio, and video. Text and images are still media, while audio and video are the continuous media. Processes of multimedia mining are given in the figure 2.3 [5].

First process is to be analyzed in this multimedia mining processes are data collection. Data collection is one of the basic process on which rest of the processes is dependent. It collects the raw audio data and then feature extraction is to be done on this data to obtain the features. After that data pre-processing are applied to the raw data for the feature selection. Data pre-processing includes data cleaning, normalization, transformation, feature selection, etc. The product of the data pre-processing is the training data. Based on these training data, we can choose a learning model. We can also move in between the tasks to improve our results.

There are mainly four types of multimedia mining:

- Text mining
- Image mining
- Video mining
- Audio mining

- **Text mining**

Much of the information is in textual form that could be data on the web or library data or electronic books, among others. The problem associated with the text data is that it is not structured as relational data. In many cases it is unstructured and in some cases it is semi structured. Semi structured data, for example, is an article that has a title, author, abstract and paragraphs. The paragraphs are not structured, while the format is structured.

The definition of the text mining is to be data mining on text data. It is all about extraction of patterns and associations previously unknown from large text databases. The difference between the information retrieval and text mining is analogous to the difference between data mining and database management system. Some of the current directions in mining unstructured data include the following:

- i. Extract data and/or metadata from the unstructured databases possibly by using tagging techniques, store the extracted data in structured databases, and apply data mining tools on the structured databases, shown in figure 2.4
- ii. Integrate data mining techniques with information retrieval tools so that appropriate data mining tools can be developed for unstructured databases as shown in figure 2.5.
- iii. Develop data mining tools to operate directly on unstructured databases is shown in the figure 2.6.

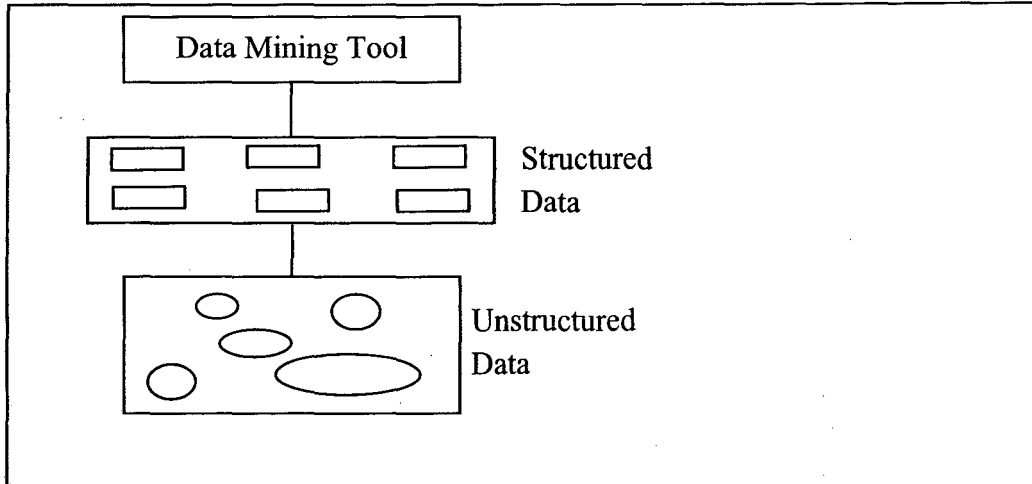


Figure 2.4: Converting Unstructured Data to Structured Data for Mining [2]

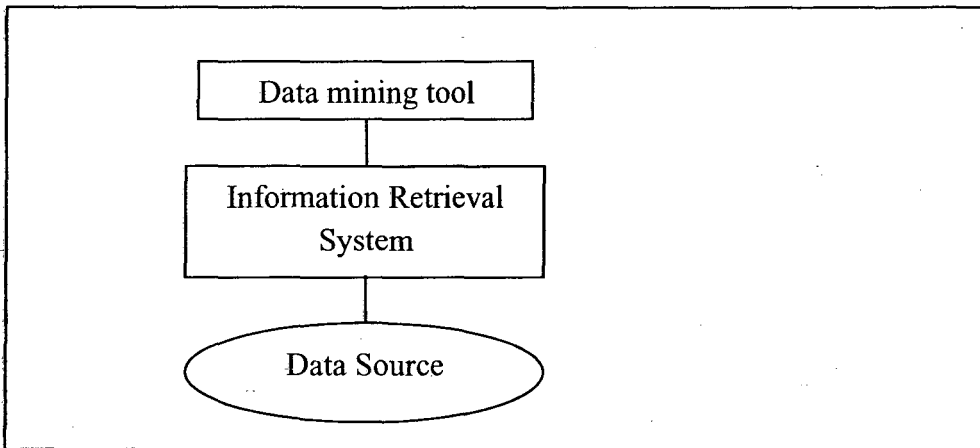


Figure 2.5: Augmenting an Information Retrieval System [2]

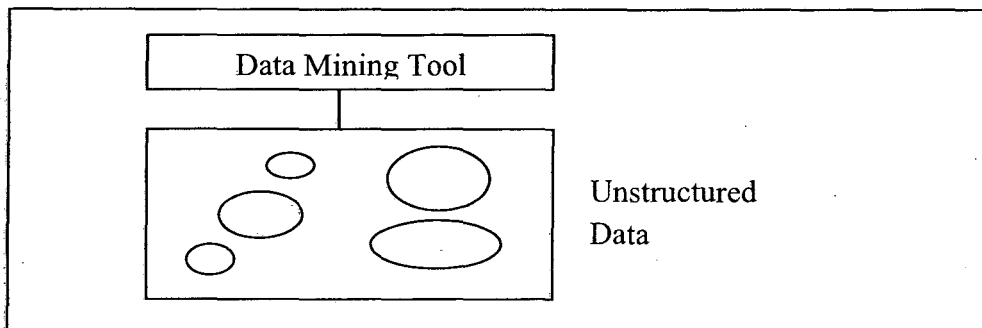


Figure 2.6: Mining Directly on Unstructured Data [2]

- **Image Mining**

Images include maps, geological structures, biological structures, and many other entities. We have Image processing applications in various domains including medical imaging for cancer detection, processing satellite images for space and intelligence applications and also handling hyper spectral images.

There is again a question arise what is the difference between an image proceeding and image mining. Image processing has dealt with areas such as detecting abnormal patterns which deviate from the norm, retrieving images by content, and pattern matching, and image mining is all about finding the unusual patterns. Initially extraction of metadata from images performed and then carries out mining on metadata.

After sometime, it was felt that image could be mined directly. The challenge then is to determine what type of mining outcome is most suitable. One could mine for association between images, cluster images, classify images as well as detect unusual patterns. So the approach is to develop templates that generate several rules about the images, and from there, apply the data mining tools to see if unusual patterns can be obtained. However, the mining tools will not tell why these patterns are unusual.

- **Video mining**

Video data mining is even more complicated than image data mining. Generally video is the collection of moving images, much like animation. The important areas include developing query and retrieval technique for video databases, including video indexing, query languages, and optimization strategies. For video mining, the handling of the image mining should be good. Finding correlation and patterns previously unknown from large video databases is video mining. So by analyzing a video clip or multiple video clips, one comes to conclusions about some unusual behavior. Another way to look at the problem is to capture the text in video format and try and make the association one would carry out with text but this time use the video data instead that is shown in figure 2.7. Converting the video mining problem to a text mining problem is reasonably well understood. However, the challenge is mining video data directly, and more importantly what we want to mine. With the emergence of the web, video mining becomes even more important.

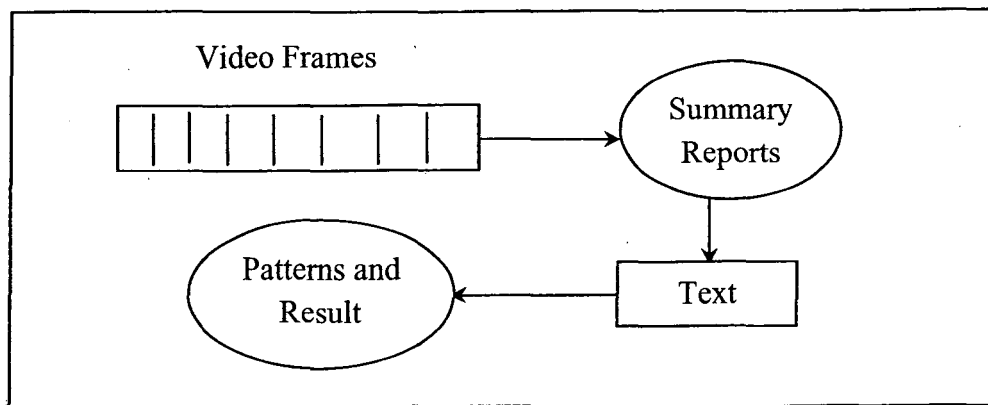


Figure 2.7: Mining Text Extracted from Video [2]

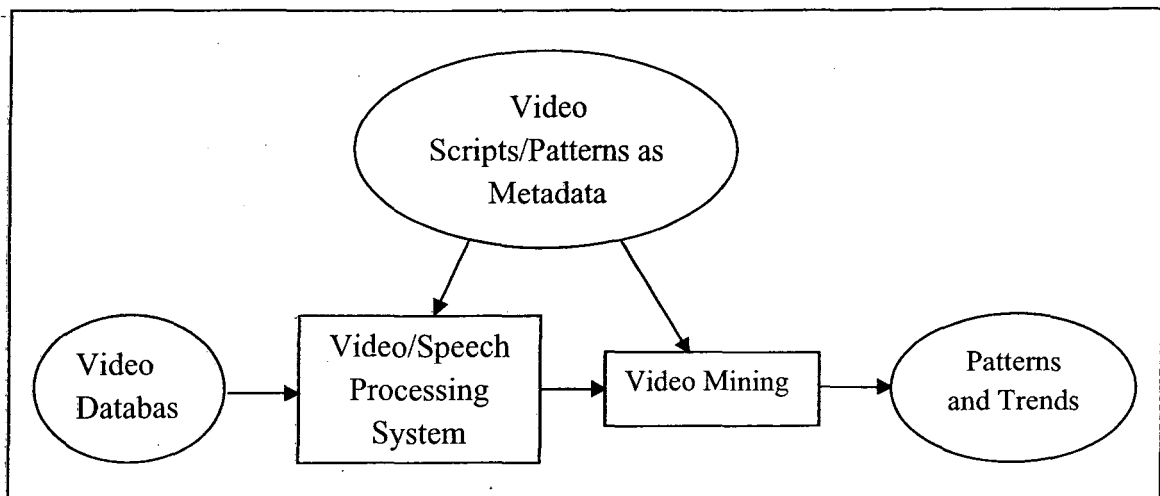


Figure 2.8: Direct Video Mining [2]

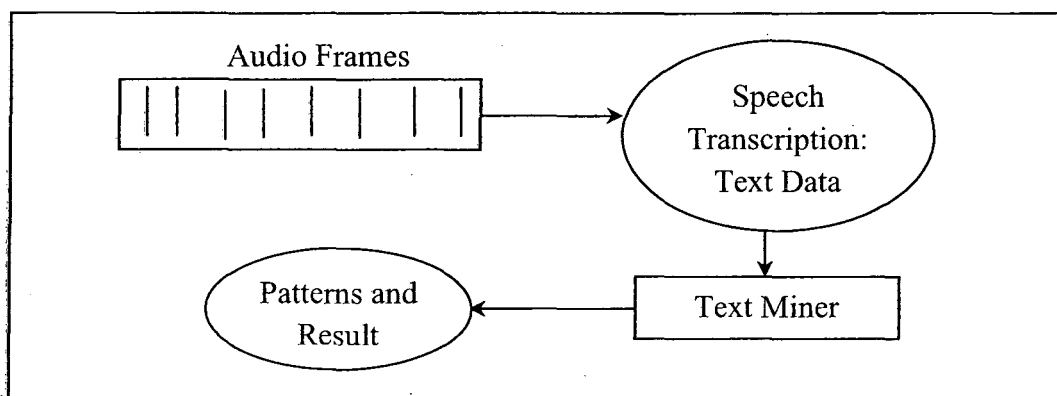


Figure 2.9: Mining Text Extracted from Audio [2]

- **Audio mining**

Since audio is a continuous media type like video, the techniques for audio information processing and mining are similar to video information retrieval and mining. Audio data could be in the form of radio, speech or spoken language that is shown in figure 2.9s. Even television news has audio data and in this case audio may have to be integrated with video and possibly text to capture the annotation and captions. To mine audio data one could convert it into text using speech transcription technique and other technique such as keyword extraction and then mine the text data. On the other hand audio data could also be mined directly by using audio information processing techniques and then mining selected audio data in figure2.10. In general, audio mining has been observed to be even more primitives than video mining.

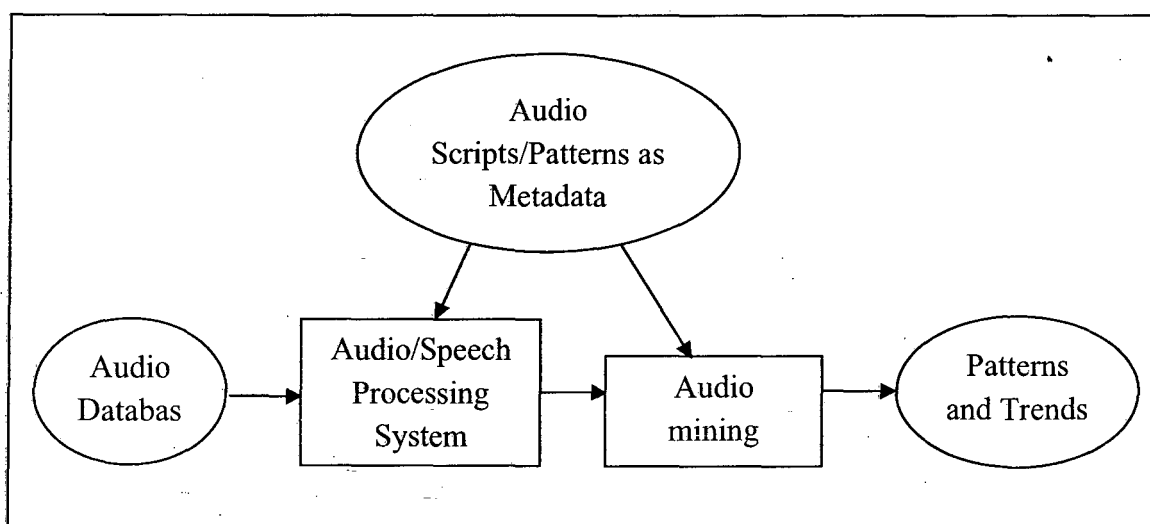


Figure 2.10: Direct Audio Mining [2]

Chapter 3

Basic Concepts

3.1 Audio Segmentation

Segmentation of the audio is the process partitioning of audio stream into the smaller units, which some times are called the indices. Within an audio data segment, the signal characteristics do not have significant variations, while the boundaries between two consecutive segments have abrupt or rapid change in the signal. The segments are usually variable-sized units. A segment of audio data roughly denotes a meaningful unit of data, which could be used as indices either directly, or as metadata from which indices could be derived by further processing. The indices represent themselves as an individual sounds [19].

3.1.1 Detecting Segmentation Boundaries

For on-line segmentation of audio data, the short-time energy function, short-time average zero-crossing rate, and short-time fundamental frequency are computed on e with incoming data. Whenever there is an abrupt change detected in any of these three features, a segment boundary is set. In the temporal curve of each feature, there are two adjoining sliding windows installed with the average amplitude computed within each window. The sliding windows proceed together with newly computed feature values, and the average amplitude within each window is updated. We compare these two values whenever there is a significant difference between them, an abrupt change is claimed to be detected at the common edge of the two windows [19] [6].

3.1.2 Classification of Segments

After segmenting boundaries are detected, each segment is classified into one of the basic audio types through the following steps as given in [19]:

3.1.2.1 Detecting Silence

The first step is to check whether the audio segment is silence or not. Silence is defined as the imperceptible audio, including unnoticeable noise and very short clicks. Generally energy function and zero-crossing rate are used to measure or detect silence. If the short-time energy functions is continuously lower than a certain set of thresholds, or if most short-time average zero-crossing rates in the segment are lower than a certain set of thresholds, then the segment is indexed as silence.

3.1.2.2 Separating Sounds with Music Component

When we are going to separate sounds and music component we have to first separate the audio segments into two categories, i.e. with or without music components mainly done by detecting continuous frequency peaks from the power spectrum that is generated by an AutoRegressive (AR) model. If there are peaks detected in consecutive power spectra which stay at about the same frequency level for a certain period of time is indexed as having music components. An index sequence is generated for each segment of sound, i.e. the index value is set to 1 if the sound is detected as having music components at that instant and to 0, otherwise. The ratio between the number of zeros in the index sequence and the total number of indices in the sequence can thus be a measurement of the sound segment as having music components or not (we call it “zero ratio”). The higher the ratio is, the less music components are contained in the sound.

We examine the zero ratio of different type of sounds, and summarize our observation below:

- i. **Speech:** Although the speech signal contains many harmonic components, the frequency peak change faster and last for a shorter time than those of music. The zero ratios for speech segments are normally above 0.95.
- ii. **Environmental Sound:** Harmonic and stable environmental sound are all indexed as having music components, while non-harmonic sounds are indexed as not having music components.
- iii. **Pure Music:** The zero ratios for all pure music are below 0.3. Indexing errors are normally come from short notes, low volume low frequency parts, non-harmonic components, and the intermissions between two notes.
- iv. **Song:** Most song segments have a zero ratio below 0.5. Those parts not detected as having music component result from peak tracks shaped like ripples (instead of lines) when the notes is long, intermission between notes, low volume/or low frequency sounds. When the ripple shaped peak track are detected and indexed as music components, the corresponding zero ratio for songs are significantly reduced.

- v. **Speech with Music Background:** When the speech is strong, the background music is normally hidden and cannot be detected. However, music components can be detected in intermission period in speech or when music becomes stronger. We make a distinction of the following two cases. For the first case, when the music is stronger or there are many intermissions in speech so that music is prominent part of the sound, the zero ratios are below 0.6. For the second case, when the music is weak while speech is strong and continuous, speech is the major component and music may be ignored. The zero ratio is higher than 0.8.

Therefore, based on a threshold for the zero ratios at about 0.7 together with some other rules, audio segments can be separated into two categories as desired. The first category contains harmonic and stable environmental sound, pure music, song, speech with the music background. For the second category, there are pure speech and non-harmonic environmental sounds.

3.1.2.3 Detecting Harmonic Environmental Sounds

The next step is to separate out environmental sound which are harmonic and stable. The temporal curve of the short-time fundamental frequency is checked. If the most part of the curve are harmonic, and the fundamental frequency is fixed at one particular value, the segment is indexed as “harmonic and unchanged”. A typical example of this type is sound touch-tone. If the fundamental frequency of a sound clip changes over time but only with several values, it is indexed as “harmonic and stable”. Example of this type includes sounds of the doorbell and the pager.

3.1.2.4 Distinguishing Pure Music

Pure music is distinguished based on properties of the averaged zero-crossing rate and the fundamental frequency. Four aspects are checked. They are the degree of being harmonic, the degree of fundamental frequency’s concentration on certain values during a period of time, the variance of zero-crossing rates. For each aspect, there is one empirical threshold is satisfied, the decision value is set to 1; otherwise it is set to a fraction between 0 and 1 according to the distance to the threshold. The four decision values are averaged with predetermined weights to

derive a total of the audio segment to be pure music. For a segment to be indexed as “pure music” the probability should be above a certain threshold and at least three of the four decision should be above 0.5.

3.1.2.5 Distinguishing Song

We extract spectral peak for sound segment of the song, speech with music background, and environment sound with music background and differentiate the three audio types based on the analysis of these tracks. Songs may be characterized by one of the three features: ripple-shaped harmonic peak tracks (due to the vibration of vocal chords), tracks which are of a longer duration compared to those in speech, and tracks which have a fundamental frequency higher than 300 Hz. Tracks are checked to see whether any of these three feature is matched. The segment is indexed as “song” if either the sum of durations where harmonic peak tracks satisfy one of the features is above a certain amount, or its comparison to the total length of the segment reached a certain ratio.

3.1.2.6 Separating Speech with Music Background and Environmental sound with Music Background

In speech with the music background, as long as the speech is strong (i.e. the pronunciations are clear and loud enough for human perception), the harmonic peak tracks of the speech signal can be detected in spite of the music components. We check the group of tracks to see whether they concentrate in the lower to middle frequency bands (with the fundamental frequency between 100 to 300 Hz) and have a lengths within a certain range. If there is duration in which the spectral peak track satisfies these criteria, the segment is indexed as “speech with music background”. The other which left is indexed as “environmental sound with the music background”.

3.1.2.7 Distinguishing Pure Speech

When distinguishing pure speech, five conditions are checked. The first one is the relation between temporal curves of the zero-crossing rate and the energy function. In speech segments, the ZCR curve has peaks for unvoiced components and troughs for voiced components, while the energy curves have peaks for voiced components and troughs for unvoiced components. Thus,

there is compensative relation between them. The second aspect is the shape of the ZCR curve. For speech, ZCR has a stable and low baseline with peaks above it. The baseline is defined as the linking line of the lowest points of troughs in the curve. The mean and the variance are calculated. The parameters and the frequency of peaks are also considered. The third and fourth aspects are the variance and the range of the amplitude of the ZCR curve, respectively. Contrary to the music segments where the variance and the range of amplitudes are normally lower than certain thresholds, a typical speech segment has a variance and range of amplitudes that are higher than certain thresholds. The fifth aspect is related to the property of the short-time fundamental frequency. As voiced component are harmonic and non-voiced are non-harmonic, speech has a percentage of harmony within a certain range. There is also a relation between the fundamental frequency curve and the energy curve. That is, harmonic parts in fundamental frequency curve correspond to peaks in the energy curve while the zero parts in the fundamental frequency curve correspond to troughs in the energy curve. A decision value, which is a fraction between 0 and 1, is defined for each of the five conditions. The weighted average of these decision values represents the possibility of the segment's being the speech.

3.1.2.8 Classifying Non-harmonic Environmental Sounds

The last step is to classify what left in the second category into one type of the non-harmonic environmental sounds are as the following. We apply following four rules.

- i. If either the energy function curve or the average zero-crossing rate curve has peaks which have approximately equal interval between neighboring peaks, the segment is indexed as "periodic or quasi-periodic". Examples for this type include sounds of clock ticks and the regular footstep.
- ii. If the percentage of harmonic parts in the fundamental frequency curve is within a certain range (lower than the threshold for music but higher than the threshold for non-harmonic sound), the segment is indexed as "harmonic and non-harmonic mixed". For example, the sound of train horn, which is harmonic, appears with a non-harmonic background.
- iii. If the frequency centroid (denoted by the average zero-crossing rate value) is within a relatively small range compared to the absolute range of the frequency distribution, the



segment is indexed as “non-harmonic and stable”. One example is the sound of bird cry, which is non-harmonic while its ZCR curve is concentrated within the range of 80-120.

- iv. If the segment does not satisfy any of the above conditions, it is indexed as “non-harmonic and irregular”. Many environmental sounds belong to this type such as the sound of thunder, earthquake and fire.

3.2 Detection of Key Audio Element

Detection of key audio elements is a step when a composite audio is temporally segmented into mono-model segments including speech, music or background noise, and then the key elements are extracted for the segments. Since speech, music, or noise are considered as a key element in semantic (middle level) discovery, by using these semantics we can also extract the high-level semantics like applause, laughter, cheer, car-braking, car-crash, gun-shot, explosion, helicopter, plane, and siren.

In [13] the Hidden Markov Model (HMM) is used for key audio effect modeling providing a natural way for modeling time-varying process. Unsupervised k-mean clustering with Bayesian Information Criterion (BIC) is performed on the training set to estimate the HMM states of each key audio effect model.

3.3 Audio Features

In [18], [11], [10] have defined the different audio features. Audio features are the characteristics of the audio data. To obtain the feature we must go through the feature extraction process. Feature extraction is the basic step for analysis of any audio stream. There have been many types of features identified based on the various methods of analysis. Descriptive features are quit difficult to extract due to the complexity of the human audio perception. There is no feature has a 100% certainty to differentiate between to different audio classes. So that combination of feature is used to achieve reasonable high classification accuracy into different categories. Some of the features are describe below:

3.3.1 Zero-Crossing Rate (ZCR)

ZCR is defined as the number of time-domain zero-crossing within a processing window. ZCR generally counts the number of the times that an audio signal crosses its zero axis [4].

It is calculated by a given formula:

$$ZCR = \frac{1}{M-1} \sum_{m=0}^{M-1} |\text{sign}(x(m)) - \text{sign}(x(m-1))|$$

Where,

M-total no of sample in a processing window,

X (m)-is the value of mth sample,

Sign is a function defined as

$$\text{sign}(a) = \begin{cases} 1 & a > 0 \\ 0 & a < 0 \end{cases}$$

This has is simple and low computational complexity. ZCR is used by most of the author to differentiate speech and music, and classification of audio into different music genres. Speech consists of voiced and unvoiced sounds. ZCR show a relationship between the frequency content of the signal. Therefore voiced and unvoiced sounds have respectively low and high zero crossing rate.

3.3.2 Short-time Energy

Short time energy is a simple feature that is broadly used in the audio classification. It is defined as the sum of a squared time domain sequence of data i.e.

$$STE = \sum_{m=0}^{M-1} x^2(m)$$

Where,

M-total no of sample in a processing window,

X (m)-is the value of mth sample.

Short time energy is the measure of the energy in a signal useful in discriminating the speech and music. Speech consists of word and mixed up with the silence which makes the variation of the short time energy value for speech higher than music.

3.3.3 Root Mean Square

Root mean square is also a measure of energy in the audio signal. It is defined as the squared root of the average of a squared signal data i.e.

$$RMS = \sqrt{\frac{1}{M} \sum_{m=0}^{M-1} x^2(m)}$$

Where,

M-total no of sample in a processing window,

x (m)-is the value of mth sample,

Similar to short time energy it is used to discriminate between speech and music.

3.3.4 High Feature Value Ratio

Before calculating high feature value ratio we have to calculate high zero crossing rate ratios. High zero crossing rate ratios is defined as the ratio of number of frames whose zero crossing rate is above 1.5-fold average zero crossing rate in a 1 second window. It is calculated by given formula:

$$HZCRR = \frac{1}{2N} \sum_{n=0}^{N-1} [\text{sign}(ZCR(n) - 1.5\text{avZCR}) + 1]$$

Where,

N- is the total number of frames,

n- frame index,

ZCR(n)- zero crossing rate at the nth frame,

avZCR- average zero crossing rate in a 1 second window,

Sign is a function defined as

$$\text{sign}(a) = \begin{cases} 1 & a > 0 \\ 0 & a < 0 \end{cases}$$

Now this value of high zero crossing rate ratio values are used as feature value in the high feature value ratio. It is defined as the ratio of number of frames whose feature value is above 1.5-fold average feature value in a processing window. It is calculated as:

$$HFVR = \frac{1}{2N} \sum_{n=0}^{N-1} [\text{sign}(FV(n) - 1.5\text{avFV}) + 1]$$

Where,

N- is the total number of frames,
n- frame index,
FV(n)- feature value at the nth frame,
avFV- average feature value in a processing window,
Sign is a function defined as

$$\text{sign}(a) = \begin{cases} 1 & a > 0 \\ 0 & a < 0 \end{cases}$$

3.3.5 Low Feature Value ratio

Similar to high feature value, before calculating low feature value we have to calculate first the low short time energy ratio. Low short time energy ratio is defined as the ratio of number of frames whose short time energy is below 0.5-fold average short time energy ratio in a 1 second window. It is calculated by:

$$LSTER = \frac{1}{2N} \sum_{n=0}^{N-1} [\text{sign}(0.5\text{avSTE} - \text{STE}(n)) + 1]$$

Where,

N- is the total number of frames,
n- frame index,
STE(n)- short time energy at the nth frame,
avSTE- average short time energy in a 1 second window,
Sign is a function defined as

$$\text{sign}(a) = \begin{cases} 1 & a > 0 \\ 0 & a < 0 \end{cases}$$

Now this value of low short time energy ratio values is used as feature value in the high feature value ratio. It is defined as the ratio of number of frames whose feature value is below 0.5-fold average feature value in a processing window. It is calculated as:

$$LFVR = \frac{1}{2N} \sum_{n=0}^{N-1} [\text{sign}(0.5\text{avFV} - \text{FV}(n)) + 1]$$

Where,

N- is the total number of frames,
n- frame index,

TE(n)- short time energy at the nth frame,

avSTE- average short time energy in a processing window,

Sign is a function defined as

$$\text{sign}(a) = \begin{cases} 1 & a > 0 \\ 0 & a < 0 \end{cases}$$

3.3.6 Spectrum Centroid

This feature generally used to classify the noise, speech and music also in different music genres. It is based on analysis of the frequency of the audio signal. Frequency spectrum is calculated by the discrete Fourier transform i.e.

$$A(n, k) = \left| \sum_{m=-\infty}^{\infty} x(m)w(nL - M)e^{-j\left(\frac{2\pi}{L}\right)km} \right|$$

Where,

k-is the frequency bin of the nth frame,

x(m)- is the input signal,

w(m)- window function,

L- window length

Spectrum Centroid is a metric of the center of gravity of the frequency power spectrum.

Spectrum Centroid is a measure that signifies if the spectrum contains a majority of high or low frequency. It is calculated as:

$$SC(n) = \frac{\sum_{k=0}^{K-1} k \cdot |A(n, k)|^2}{\sum_{k=0}^{K-1} |A(n, k)|^2}$$

Where,

K- is the order of the discrete Fourier transform,

k- frequency bin for nth frame of the signal,

3.3.7 Spectrum Spread

Spectrum spread is the measure that signifies if the power spectrum is concentrated around the Centroid or if it is spread over the spectrum. Spectrum spread is useful in discriminating music and speech because music is the broad mixture of the frequencies whereas the speech contains limited number of frequencies, also used for classifying different music genres. It is calculated as:

$$SS(n) = \sqrt{\frac{\sum_{k=0}^{K-1} (k - SC)^2 \cdot |A(n, k)|^2}{\sum_{k=0}^{K-1} |A(n, k)|^2}}$$

Where,

K- is the order of the discrete Fourier transform,

k- frequency bin for nth frame of the signal,

A(n,k)- discrete Fourier Transform of the nth frame of a signal.

3.3.8 Spectral Flux

Spectral flux is often called as delta spectrum. It measures frame to frame spectral difference. It is defined as the average variation value of spectrum between the two adjacent frames in a processing window. Generally this feature is used to discriminate the speech and music, and also used to distinguish the music and environmental sound. It is calculated as:

$$SF = \frac{1}{(N-1)(K-1)} \sum_{n=1}^{N-1} \sum_{k=1}^{K-1} [\log(A(n, k) + \delta) - \log(A(n-1, k) + \delta)]^2$$

Where,

N- is the total number of frames,

K- is the order of the discrete Fourier transform,

δ - is a very small value used to avoid calculation overflow

A(n,k)- discrete Fourier Transform of the nth frame of a signal.

3.3.9 Spectral Rolloff Frequency

Spectral rolloff frequency is defined as measure of how high in the frequency spectrum a certain part of energy lies. Speech has less spectral rolloff frequency rather than in comparison to the music. It is calculated as:

$$SRF(n) = \max (h \left| \sum_{k=0}^h A(n, k) \right| < TH. \sum_{k=0}^{K-1} |A(n, k)|^2)$$

Where,

N- is the total number of frames,

K- is the order of the discrete Fourier transform,

TH- is a threshold set to the value of 0.92

A(n,k)- discrete Fourier Transform of the nth frame of a signal.

3.3.10 Mel Frequency Cepstral Coefficient (MFCC)

They are the compact representation of spectrum. They are represented according to the information of the audio signal. It is modeled to capture the relevant parts of the auditory system. Generally it is the inverse of the Fourier transform. Actually there are thirteen coefficients are supposed to be found in representation of the speech. It is calculated as:

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

Where, f is the frequency of an audio signal.

3.4 Annotation

Annotation is representation of metadata and contents (extracting element by segmentation) of audio data. Annotation is based on the features (low level) extracted from the segments of the audio stream. For a given audio stream data each of the features could in turn be described by its characteristic vector. Examples of such features are pitch, pitch duration . . . (for melody, tonality), timing of a beat, strength of a beat . . . (for rhythm), etc. Therefore it may be possible to analyze an audio stream after performing its annotation with a computed value for each of these characteristics [7].

3.5 Classification Algorithm/Technique

3.5.1 Gaussian Classifier

Gaussian classifier is an example of the parametric classifier. It is based on the assumption that feature vector of each class obey a multidimensional Gaussian distribution. Estimate of the parameters (mean and covariance) of the Gaussian probability density function of each class are

computed using the training data at the training stage. The input vector is mapped to the class with the largest likelihood at the classification stage.

In k-dimension, the probability density function is expressed as:

$$p(X) = \frac{1}{(2\pi)^{k/2} |\Sigma|^{1/2}} \exp \left\{ -\frac{1}{2} (X - \mu) \Sigma^{-1} (X - \mu) \right\}$$

Where,

Σ -is the $k \times k$ covariance matrix and

μ - is a mean k-dimensional vector.

3.5.2 The K-Nearest Neighbor Classifier

It is an example of non-parametric classifier. In this classifier, each input feature vector to be classified, a search is made to find the location of k-nearest training examples, and then assign the input to the class having the largest members in this location. Euclidian distance is used as a metric to measure the neighborhood in this classifier. The Euclidian distance between feature vectors $X = \{x_1, x_2, \dots, x_n\}$ and $Y = \{y_1, y_2, \dots, y_n\}$ is calculated as:

$$d = \sqrt{\sum_{i=1}^n (X_i - Y_i)^2}$$

Disadvantages of the K-Nearest Neighbor classifier are:

1. Whenever a new feature vector is classified need the entire feature vector of all training data and hence required a large storage requirement.
2. The classifying time is more in comparison to other classifiers.

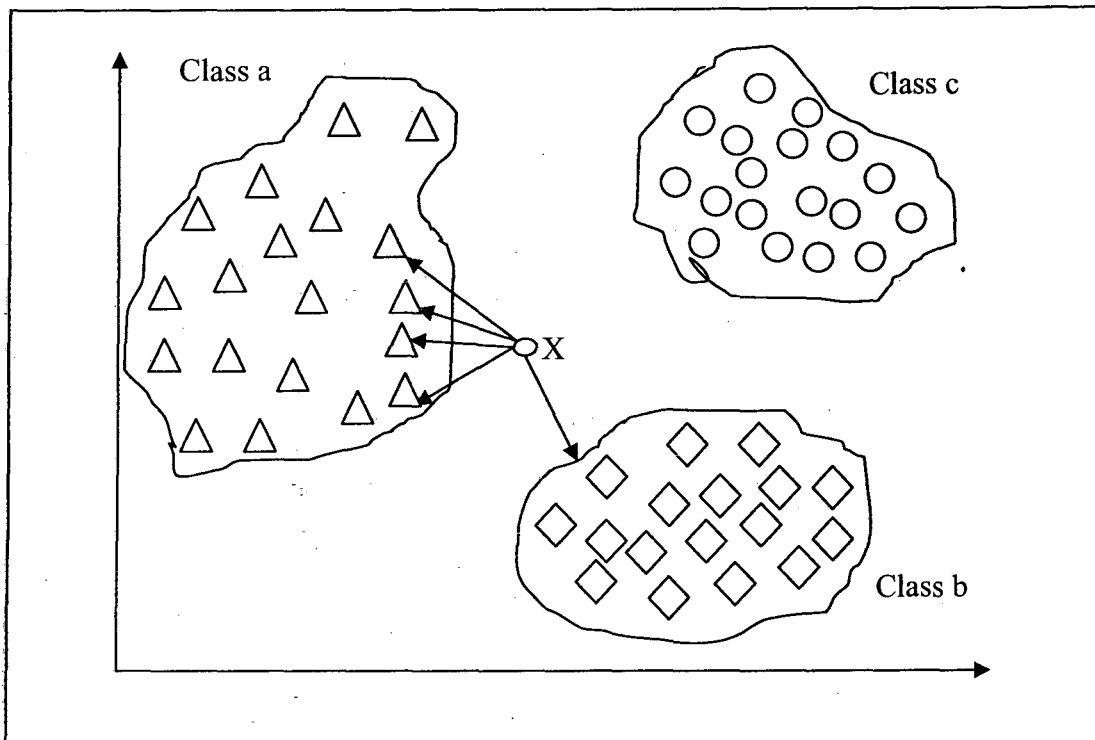


Figure 3.1: K-Nearest Neighborhood (K=5) [1]

3.5.3 C4.5 Classifier

C4.5 is an algorithm to generate the decision tree developed by Ross Quinlan. C4.5 is an extension of the ID3 algorithm of Ross Quinlan. C4.5 builds a decision tree from a training data as it was done by the ID3 algorithm by using the concept of information entropy. The training set contains the set of samples $S = s_1, s_2, s_3, \dots$ in which each sample $s_i = x_1, x_2, x_3, \dots$ is a vector where each vector x_i represents attribute or the feature of the samples. The training data is augmented with a vector $C = c_1, c_2, c_3, \dots$ where c_1, c_2, \dots represents the class to which each class belongs to.

Information entropy is the measure of the uncertainty associated with a random variable. The information entropy of a discrete random variable X , that can take on possible values $\{x_1, \dots, x_n\}$ is:

$$H(X) = E(I(X)) = \sum_{i=1}^n p(x_i) \log_2(1/(p(x_i)))$$

Where,

$I(X)$ – is the information content,

$P(x_i)$ – is the probability mass function of

The java implementation of the C4.5 decision tree is known as the j48 classifier which is used in our experiments. A decision tree is a predictive machine learning model that decides the target value i.e. the dependent variable of a new sample based on the various attribute values of the available data. The internal node of a decision tree denote the different attributes, and the branch between the nodes denote the possible values that these attribute can have in the observed samples, while the terminal node tell us the final values that is the classification data of the dependent variable [25].

3.5.4 Bayesian Classifier

Bayesian classifier is the simple probabilistic classifier based on Bayes' theorem with independent assumption. Bayesian classifier, depending upon the precise nature of probability model, can be trained very efficiently in a supervised learning. It is suited to those which have high dimensionality of input. Despite of simple probabilistic classifier, Bayesian classifier can often outperform more sophisticated method [25][8].

Bayesian classifier is generally based on Bayes' theorem which relates the conditional and marginal probabilities of stochastic events A and B. stochastic events or random events that are the counterpart of a deterministic process considered in probability theory. So, Bayes' theorem is given as:.

$$P(A/B) = \frac{P(B/A)P(A)}{P(B)} \propto L(A/B)P(A)$$

Where,

$L(A/B)$ – is the likelihood of A given fixed B,

$P(A)$ - is the prior or marginal probability of A,

$P(A/B)$ – is the conditional probability or posterior probability of A depending upon the specified value of B,

$P(B)$ -is the prior or marginal probability of B,

3.5.5 Hidden Markov Model (HMM)

A Hidden Markov Model is a statistical model in which a system is assumed to be a Markov process with unknown parameters, and challenge is to determine the hidden parameters from the observable parameters. The extracted model parameter can then be used to perform further analysis. A HMM can be considered as the simplest dynamic Bayesian network.

In a regular Markov Model, the state is directly visible to the observer, and therefore the state transition probabilities are the only parameters. In HMM, the state is not directly visible, but variable influenced by the state are visible. So those, the sequence of tokens give the information about the sequence of states [25].

3.5.6 Multi Layer Perception (MLP) Neural Network

It is actually a multilayer feed forward Neural Network. The leaning algorithm i.e. the back propagation algorithm is performed on this neural network. MLP Neural Network is organized in the layers that are made up of a number of interconnection nodes which contain an activation function. Patterns are represented in the network via the input layer, which communicate to one or more hidden layers where the actual processing is done via a system of weighted connections. The hidden layer is then link to an output layer where the output comes [9].

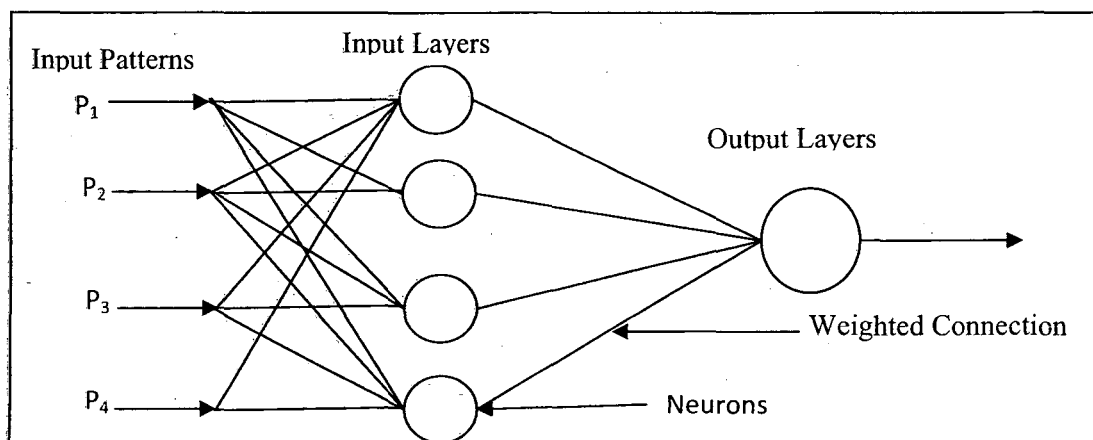


Figure 3.2: A Feed Forward Neural Network Structure [9]

Chapter 4

Proposed Method

4.1 Framework for Audio Classification

The classification problem considered in this dissertation is a two class problem. Therefore the audio database may contain labeled audio data of any two classes, such as speech-music, speech-environmental sound, vocal-instrumental, silence-speech, song-environmental sound etc.

A framework considered in this dissertation to classify audio data has been presented in figure 4.1. The framework comprises two phases namely audio data representation phase and classifier modeling phase. In figure 4.1 (a), the Audio Representation, audio data is the input. Based on the segmenting boundaries of the audio data the features are extracted for each segment to generate the feature vector for each audio data file and this may generate the feature matrix for a set of audio data files. In phase 2, feature matrix obtained from phase 1 is the input to the audio data classifier may be generated using a suitable algorithm to model a classifier.

The features of audio data have been discussed in detail in Chapter 3. Considering the format of raw data being waveform, in this dissertation, the following five time-domain feature have been identified:

- i. Zero Crossing Rate (ZCR)
- ii. Short-Time Energy (STE)
- iii. Root Mean Square (RMS)
- iv. Low Feature Value Ratio (LFVR)
- v. High Feature Value Ratio (HFVR)

Each component of a feature vector is the average value of the corresponding feature of all the segments of the audio data file. Thus this feature vector is the annotation of the audio data file. The numeric values of the features have been used to obtain the feature vector. This feature matrix of set of audio files serves as a input database for the classifier, which classify them in audio classes. In phase 2, two classifiers are used for the classification purpose, first is the Bayesian Classifier and the second is the Decision Tree (C4.5).

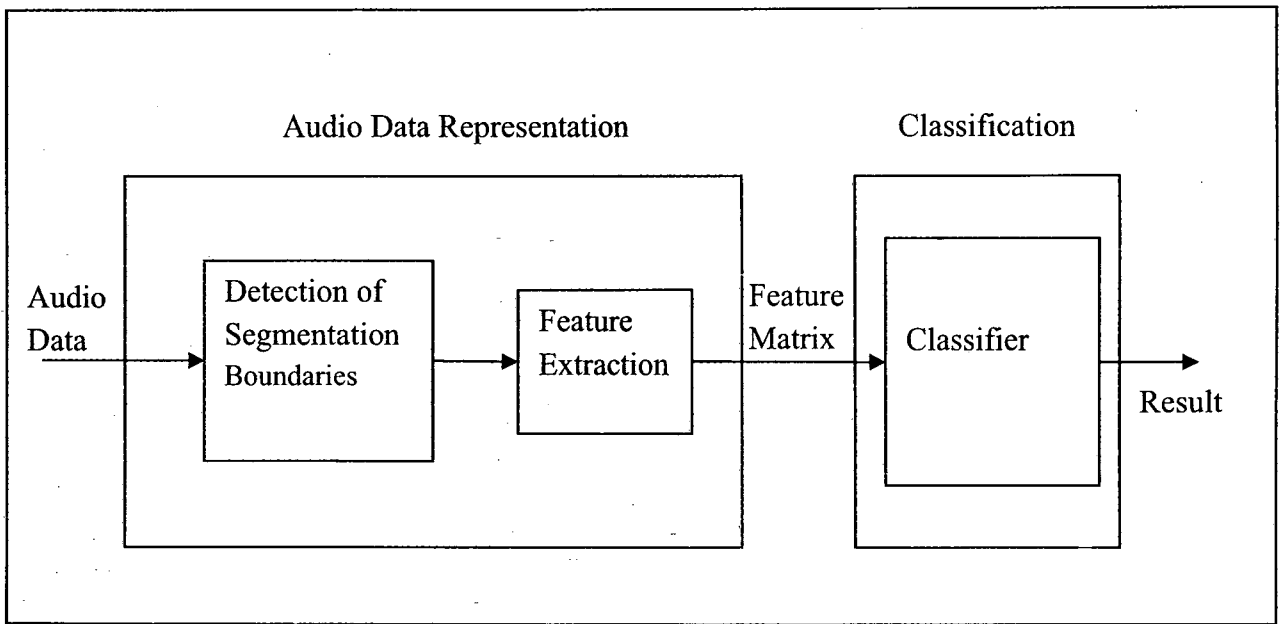


Figure 4.1 (a): Framework for Audio data Classification

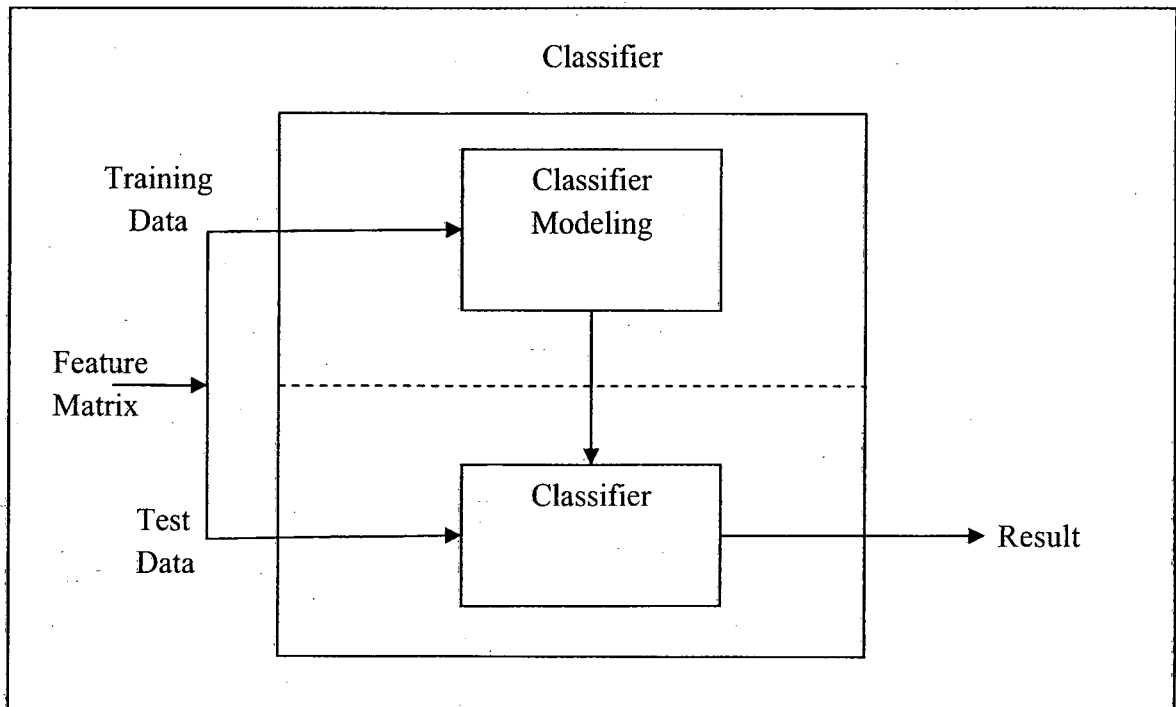


Figure 4.1 (b): Audio Data Classification

4.1.1 Audio Data Representation

An audio data may consist of a number of segments. The concept segmentation and segments have been recaptured in Chapter 3. The segments are annotated by low level features such as zero crossing rate, root mean square, spectral centroid, bandwidth, spectral roll-off frequency, band energy ratio, delta spectrum magnitude, pitch, pitch strength, etc. an audio data may either be represented using segment level features or some other features extracted on the basis of the segment level features. The above list of the feature may be used to annotate an audio data on the basis of segment level features. Thus corresponding each sample audio data has a 5-tuple such as (ZCR, STE, RMS, LFVR, HFVR) may represent the data. As the problem undertaken in this dissertation auctions to classification, to each data samples, the class label also associated. Thus an audio file such as xyz.wav may be represented by a 6-tuple i.e.

xyz.wav: (ZCR, STE, RMS, LFVR, HFVR, class-label)

4.1.2 Classification

Classification is an instance of supervised learning, where a model or pattern is learnt to predict the data instances. Classification is used to predict the category of the categorical data by building a model based on some predictable variables. In section 4.1.1, the feature matrix has been generated by annotating the audio data. The feature matrix is used as a predictable variable on the basis of which a model may be built. The feature matrix has been divided into two data set i.e. training data set and test data set. In figure 4.1 (b), the training data set is the input to the classifier model. The result obtained by the classifier model is the classification model on the basis of that test data set is classified into the class labels. The algorithms of two classifiers are described here.

1. Naive Bayesian Classifier

Bayesian classifiers are the statistical classifiers. They can predict class membership probabilities, such as the probability that a given sample belong to a particular class. Naïve Bayesian classifier assumes that the effect of an attribute on a given class is independent of the values of the other attributes, which is called as the class conditional independence [8].

Bayesian classifier is generally based on Bayes' theorem which relates the conditional and marginal probabilities of stochastic events A and B. Stochastic events or random events that are the counterpart of a deterministic process considered in probability theory. So, Bayes' theorem is given as:

$$P(A/B) = \frac{P(B/A)P(A)}{P(B)}$$

Where,

P (A) - is the prior or marginal probability of A,

P (A/B) – is the conditional probability or posterior probability of A depending upon the specified value of B,

P (B) - is the prior or marginal probability of B,

Naïve Bayesian classifier works on following methods:

- i. Each data sample is represented by an n-dimensional feature vector, $X=(x_1, x_2, \dots, x_n)$, depicting n measurements made on the sample from n attributes, respectively, $A_1, A_2, A_3, \dots, A_n$.
- ii. Suppose that there are m classes, C_1, C_2, \dots, C_m . Given an unknown data sample, X, the classifier will predict that the X belongs to the class having the highest posterior probability, conditioned on X. that is, the naïve Bayesian classifier assigns an unknown sample X to the class C_i if and only if

$$P(C_i | X) > P(C_j | X) \text{ for } 1 \leq j \leq m, j \neq i.$$

The class C_j for which $P(C_j | X)$ is maximized is called the maximum posteriori hypothesis.

By Bayes Theorem,

$$P(C_i | X) = \frac{P(X | C_i)P(C_i)}{P(X)}$$

- iii. As P(X) is constant for all classes, only $P(X | C_i)P(C_i)$ need be maximized. If the class prior probabilities are not known, then it is commonly assumed that the classes are equally likely, that is, $P(C_1) = P(C_2) = \dots = P(C_m)$, and therefore maximize $P(X | C_i)$. Otherwise $P(X | C_i)P(C_i)$ need to be maximized.
- iv. Given data sets with many attributes, it would be extremely computationally expensive to compute $P(X | C_i)$. In order to reduce the computation in evaluating $P(X | C_i)$, the naïve assumption of class conditional independence is made. This presumes that the values of the

attributes are conditionally independent of one another, given the class label of the sample, that is, there are no dependence relationships among the attributes. Thus,

$$P(X|C_i) = \prod_{k=1}^n P(x_k|C_i)$$

The probabilities $P(x_1|C_i)$, $P(x_2|C_i)$, ..., $P(x_n|C_i)$ can be estimated from the training samples, where

1. If A_k is categorical, then

$$P(x_k|C_i) = \frac{s_{ik}}{s_i}$$

Where s_{ik} is the number of training samples of class C_i having the value x_k for A_k , and s_i is the number of training samples belonging to C_i .

2. If A_k is continuous-valued, then the attribute is typically assumed to have a Gaussian distribution so that

$$P(x_k|C_i) = g(x_k, \mu_{C_i}, \sigma_{C_i}) = \frac{1}{\sqrt{2\pi\sigma_{C_i}}} e^{-\frac{(x_k - \mu_{C_i})^2}{2\sigma_{C_i}^2}}$$

Where $g(x_k, \mu_{C_i}, \sigma_{C_i})$ is the Gaussian (normal) density function for attribute A_k while μ_{C_i} and σ_{C_i} are the mean and standard deviation, respectively, given the values for attribute A_k for training samples of class C_i .

v. In order to classify an unknown sample X , $P(X|C_i)P(C_i)$ is evaluated for each class C_i . Sample X is then assigned to the class C_i if and only if

$$P(X|C_i)P(C_i) > P(X|C_j)P(C_j) \text{ for } 1 \leq j \leq m, j \neq i.$$

In other words, it is assigned to the class C_i for which $P(X|C_i)P(C_i)$ is the maximum [8].

2. C4.5 Decision Tree classifier

A decision tree is a flow-chart-like tree structure, where each internal node denotes a test on an attribute, each branch represents an outcome of the test, and leaf nodes represent classes or class distribution.

The algorithm of generating a decision tree is given as:

Algorithm: Generate_decision_tree. Generate a decision tree from given training data.

Input: the training samples, represented by discrete valued attributes; the set of candidate attributes, attribute-list.

Output: A decision tree.

Methods:

1. Create a node N;
2. **If** samples are all of the same class, C **then**
3. return N as a leaf node labeled with the class C;
4. **if** attribute list is empty **then**
5. return N as a leaf node labeled with the most common class in samples; //majority voting
6. select test attribute, the attribute among attribute list with the highest information gain;
7. label node N with test-attribute;
8. **for each** known value a_i **of** test-attribute // partition the samples
9. grow a branch from node N for the condition test attribute = a_i ;
10. let s_i be the set of samples in a samples for which test attribute = a_i ; //a partition
11. **if** s_i is empty **then**
12. attach a leaf labeled with the most common class in samples;
13. **else** attach the node returned by Generation_decision_tree (s_i , attribute-list-test-attribute);

The J48 classifier follows the simple algorithm, for a new item set it first create a decision tree based on attribute values of the available training data. So whenever a set of items are encountered, it identifies the attribute that discriminate the various instances most clearly. So that these features value i.e. the discriminating attribute is able to tell about the data instances and to classify the data according to their highest information gain. It is generally used for the numeric attributes values and having nominal classes [8].

4.2 Performance Evaluation

i. Cross Validation

It is a method of estimating true error of a model. Cross validation is a method for better approximating the error that might occur while building a model for the new or unseen data. It is also used to evaluate a model in deciding which algorithm to deploy for learning, when choosing from amongst a number of learning algorithm [24]. It is preferred over the other method when a

plenty of data are available. There is a rule that there is good result obtained if the whole data is divide into a training set (66%) and test set (33%).

Here we are using the k-fold cross validation method that split the data into the k subset of the same size. The learning scheme then trained k times; they leave one of the subset from the training each time, and used to compute error estimate from left out subset. The average of the k error estimates can be taken as estimate of the generalization of a given model or it can be used for model selection.

ii. Confusion Matrix

Confusion matrix contains the information about the actual and prediction classification done by the classification system. Performance of such system is commonly evaluated using the data in matrix. As we are using two class classifying problem, so that for two class problem there are the table showing the confusion matrix [23].

Table 4.1: Confusion Matrix [23]

		Prediction	
		Negative	Positive
Actual	Negative	a	b
	Positive	c	d

Where,

- a is the number of correct predictions that an instance is negative
- b is the number of incorrect prediction that an instance is positive
- c is the number of incorrect prediction that an instance is negative
- d is the number of correct prediction that an instance is positive

Some of the terms related to the evaluation parameter for the two class classifying problem are:

1. Accuracy (AC)

The accuracy is the prediction of the total number of predictions that were correct. It is calculated as:

$$AC = \frac{a + d}{a + b + c + d}$$

2. Precision (P)

Precision is the proportion of the predictive positive cases that were correct, calculated as:

$$P = \frac{a}{b + d}$$

3. True Positive Rate (TPR)

True positive rate is the proportion of the positive cases that were correctly identified calculated by the equation:

$$TPR = \frac{a}{c + d}$$

4. False Positive Rate (FPR)

False positive rate is the proportion of the negative cases that were incorrectly classified as positive is calculated as:

$$FPR = \frac{b}{a + b}$$

Chapter 5

Experiments

&

Results

5.1 Experimental Objective

The objective of the experiments is given set of audio data in waveform belonging to two classes, to model an efficient classifier and to classify the audio data into the respective two classes. In order to achieve this initially the feature matrix of all the audio data in .wav file is generated in phase 1 by extracting the time domain features.

For classification of the audio data set using efficient classifier corresponding the two classification techniques described in chapter 3, Naïve Bayesian classifier and C4.5 classifier are modeled. The feature matrix of the set of audio data has been split into training and test sets considering the four ratios 60:40, 65:35, 70:30, and 75:25.

The performances of the classifiers generated by the two algorithms are observed to identify the better of the two models.

5.2 System Characteristics

All Experiments have performed on the system with the following configuration:

- **Processor:** Intel (R) Pentium (R) 4 CPU 2.80GHz
- **Memory:** 256 MB RAM
- **Windows Dir:** C:\WINDOWS
- **Machine name:** SCSS108
- **Operating System:** Microsoft Windows XP Professional Version 2002 Service Pack 2
- **Language:** English
- **System Manufacturer:** Acer Power
- **BIOS:** Default System BIOS
- **Software:** Matlab and Weka 3.5.6

5.3 Databases

The audio files used for the experiments in this dissertation are collected from Music Technology Group website containing a large collection of audio data. Since the classification problem considered in this work is a 2 class problem, the samples collected from repository were either speech or music. A total of 45 samples have been used for the experiments. Such data file is a wave audio file of length 3 seconds. The feature matrix of the 45 .wav file is prepared in ARFF format for input to the classification phase i.e. Phase 2.

5.4 Design of Experiments

Generally there are three stages in which the whole experiments are divided. In first stage the whole database has been split into two sets, partition based on random selection. The next stage is modeling stage where using the two techniques the classifiers are generated. In this stage, for estimation of the specific models the experiments are designed to produce models of two categories using Cross validation method and without using Cross Validation method. Thus the comparison of the two algorithms is based on the values of the evaluation parameter accuracy and precision. In order to compute the true positive (TP), false positive (FP), true negative (TN), false negative (FN), are observed for each execution. The two algorithms are compared on their performance on the training as well as test data.

Training and Test Data

For the experiment the dataset is partitioned into training-test dataset in four different ratios by sampling with replacement. The partition in 60:40 ratios is denoted by D_1 , 65:35 ratios is denoted by D_2 , 70:30 ratios is denoted by D_3 and 75:25 ratios is denoted by D_4 . Each of these has been saved separately to execute the experiments in the subsequent state.

Modeling

In the modeling phase for estimation of the performance of a classifier by using cross validation 10-fold cross validation was followed. In each of the i^{th} iteration for $1 \leq i \leq 10$, the remaining 9 folds were used to generate the classifiers while the i^{th} fold was used as the validation data. The results corresponding to the i^{th} fold for all evaluation parameters were noted. Finally for each

partition D_1 , D_2 , D_3 , D_4 the average results were computed. The classifiers were also used to classify the data corresponding to the partition.

Modeling without using Cross validation

Similarly for all the four partitions under this category the entire data was used to model the classifier and then executed to classify the entire training set. The model was also executed on the test data and all the execution parameters were observed corresponding the training and test data.

5.5 Results and Analysis

The results of the experiments carried out as described in the previous section are presented in the Tables 5.1, 5.2, 5.3, 5.4. For the learning scheme using cross validation method the results presented are the average value of the performances of the 10 results recorded for each of the 10 folds. The Average accuracy of the two classification models Naïve Bayesian and C 4.5, for validation and test data are presented for all the four data partitions namely, D_1 , D_2 , D_3 , D_4 , in Table 5.1. The values of other evaluation parameters namely, precision, TPR and FPR for the experiments with cross validation are presented in Table 5.2.

The results of the experiments without using cross validation presenting the accuracy of the two classifiers both for training as well as test data is presented in Table 5.3, while the precision, TPR and FPR for the two classifiers corresponding the data partition are presented in Table 5.4.

It may be noted that in the table 5.1, the average accuracy of the training data of data partitions D_2 , D_3 , D_4 , are the same for the two classification models but the value differs only for the training data of the data partition D_1 . The average accuracy for Naïve Bayesian classifier in this case is 4% more than the average accuracy of C4.5 classifier. For the test data of all data partitions both of the classifiers have the same result. The values of average precision for the Naïve Bayesian model as presented in Table 5.2 is 0.95875, the value of average TPR is 0.99 and the value for the average FPR is 0.1445 for the training data. Similarly the values of average precision for the decision tree classifier using C4.5 model is 0.95725, the value of average TPR is 0.95725 and the value for the average FPR is 0.1445. However, for the test data for both the

models precision, TPR and FPR are 1, 1 And 0 respectively. Thus it is visible that the performance of Naïve Bayesian is marginally better than the performance of the C4.5 model when using Cross Validation method.

Table 5.1: Classifying Audio Data Using Cross Validation

Train-Test Partition of Dataset	Bayesian Classifier (Average Accuracy in %)		Decision Tree (Average Accuracy in %)	
	Training Data	Test Data	Training Data	Test Data
D1	96.2963	94.4444	92.5926	94.4444
D2	93.1034	100	93.1034	100
D3	93.75	100	93.75	100
D4	94.1176	81.8182	94.1176	81.8182

Table 5.2: Performance Evaluation Parameter for Bayesian and Decision Tree Classifiers using cross validation

Data Sets	Bayesian Classifier						Decision Tree Classifier					
	Training data			Test data			Training data			Test data		
	Preci- sion	T.P.R	F.P.R	Preci- sion	T.P.R	F.P.R	Preci- sion	T.P.R	F.P.R	Preci- sion	T.P.R	F.P.R
D1	0.955	1	0.167	1	1	0	0.952	0.952	0.167	1	1	0
D2	0.957	1	0.143	1	1	0	0.955	0.955	0.143	1	1	0
D3	0.96	0.96	0.143	1	1	0	0.96	0.96	0.143	1	1	0
D4	0.963	1	0.125	1	1	0	0.962	0.962	0.125	1	1	0

Similarly in table 5.3, the average accuracy of the training data of data partition D₁, Naïve Bayesian has 4% more accuracy than C4.5 classifier. In case of training data of partition D₂, Naïve Bayesian has 3.7% more accuracy than C4.5 classifier. For the training data of partitions D₃ and D₄, Naïve Bayesian has 3.3% and 3.1% more accuracy than C4.5 classifier respectively, while test data of all datasets, both the classifiers have the same result. The values of average precision for the Naïve Bayesian model as presented in Table 5.4 is 0.95925, the value of average TPR is 1 and the value for the average FPR is 0.1445 for the training data. Similarly the values of average precision for the decision tree classifier using C4.5 model is 0.95925, the value of average TPR is 1 and the value for the average FPR is 0.1445. However, for the test data for both the models precision, TPR and FPR are 1, 1 And 0 respectively. Thus it is visible that the performance of Naïve Bayesian is same as the performance of the C4.5 model when Cross Validation method is not used.

Table 5.3: Classifying Audio Data without Using Cross Validation

Train-Test Partition of Dataset	Bayesian Classifier (Average Accuracy in %)		Decision Tree (Average Accuracy in %)	
	Training Data	Test Data	Training Data	Test Data
D1	96.2963	94.4444	92.5926	94.4444
D2	96.5517	100	93.1034	100
D3	96.875	100	93.75	100
D4	97.0588	81.8182	94.1176	81.8182

Table 5.4: Performance Evaluation Parameter for Bayesian and Decision Tree

Classifiers without using cross validation

Data Sets	Bayesian Classifier						Decision Tree Classifier					
	Training data			Test data			Training data			Test data		
	Preci- sion	T.P.R	F.P.R	Preci- sion	T.P.R	F.P.R	Preci- sion	T.P.R	F.P.R	Preci- sion	T.P.R	F.P.R
D1	0.955	1	0.167	1	1	0	0.955	1	0.167	1	1	0
D2	0.957	1	0.143	1	1	0	0.957	1	0.143	1	1	0
D3	0.962	1	0.143	1	1	0	0.962	1	0.143	1	1	0
D4	0.963	1	0.125	1	1	0	0.963	1	0.125	1	1	0

Chapter 6

Conclusion & Future Work

CONCLUSION

The dissertation contains a brief overview of the frameworks available in the literature for audio data analysis in Chapter 2. Some important processes such as segmentation, feature extraction of the segments and a list of features of audio data have been reviewed in Chapter 3. In the Chapter 4, the proposed method comprising the annotation based audio data representation and classification is described followed by the details of the experiments carried out on a dataset of audio data files. The results and the analysis of the results have been presented in the Chapter 5.

The problem of classification of audio data in this dissertation has been addressed by a two phased framework. In the first phase extraction of annotation based features namely; zero crossing rate (ZCR), short-time energy (STE), root mean square (RMS), low feature value ratio (LFVR), and high feature value ratio (HFVR) has been performed. The second phase is the classification phase where two classification algorithms namely Naïve Bayesian and C4.5 to induce Decision Tree classifier have been implemented. The performance of the two classifiers has been compared based on the performance parameters accuracy, true positive rate, false positive rate, and precision. The experiments were carried out on a small database of 45 audio data files. The results of the two classifiers have been observed to be satisfactory. The results of the experiments carried out indicate that Naïve Bayesian classifier is a better suited model for the classification of audio data using the above mentioned features. However the decision tree classifiers are more interpretable than the Bayesian classifiers and that the difference of the performance is 4%. Therefore the user may choose either of the classification algorithms for the classification of the audio data. Thus it is inferred that the classification of audio data is possible by a classification model using the annotation based features.

Future Work

Other classification models can also be investigated to model the best suited classifier for audio data. For better classification result, more features for audio data can be used. The database can be improved by taking larger data samples with the different audio data format such as mp3, mpeg, avi, etc. It may be possible that audio data can be classified into more than two audio class problems.

References

1. Abdillahi Hussein Omar: **Audio Segmentation and Classification**; Master's Thesis, Informaics and Mathematical Modeling Publisher, Technical University of Denmark, February 16th, 2005
2. Bhavani Thuraisingham : **Data Mining; Technologies, Techniques, Tools and Trends**: CRC Publication 1999.
3. David Bruce Gerhard: **Computationally Measurable Temporal Difference between Speech and Song**, Ph.D Thesis, University of Manitoba, April, 2003.
4. David Gerhard (2000) **Audio Signal Classification: An Overview**. Canadian Artificial Intelligence, 4-6, Winter 2000.
5. Davis Pan: **A Tutorial on Mpeg/Audio Compression**, IEEE Multimedia Publication, Volume 2, Pages (60-74), Issue 2, Summer 1995 Issues.
6. Dom Kimber, Lynn Wilox: **Acoustic Segmentation for Audio Browsers**; Xerox PARC, Palo Alto, FX Palo Alto Laboratory, Palo Alto, CA 9304, 1996.
7. George Tzaetakis, Perry Cook: **Multi-feature Audio Segmentation for Browsing and Annotation**; IEEE Multimedia Publication, New York, Pages (103-106), October 17-20th, 1999
8. Jiawei Han and Micheline Kamber: **Data Mining: Concepts and Techniques**; Morgan Kaufmann Publishers, 2001
9. Kashif Saeed Khan, Wasfi G. Al-Khatib, Muhammad Moinuddin: **Automatic Classification of Speech and Music using Neural Network**; ACM Multimedia Transaction, November 13, pages (94-99), 2004.
10. Klapuri: **Audio Signal Classification**; ISMIR Graduate School, October 4th -9th, 2004
11. Lie Lu, Hao Jiang and Hong Jiang Zhang: **A Robust Audio Classification and Segmentation Method**, Microsoft research, China, ACM Multimedia Transaction, International Multimedia Conference; Volume 9, pages (203-211), 2001.
12. Noris Mohd Norowi, Shyamala Doraisamy, Rahmita Wirza: **Factors Affecting Automatic Genre Classification**: An Investigation Incorporating Non-Western Musical

- Forms; Faculty of Computer Science and Information Technology University Putra Malaysia 43400, Selangor, MALAYSIA,2005.
13. Regunathan Radhakrishnan, Ajay Divakaran and Paris Smaragdis: **Audio Analysis for Surveillance Application**:IEEE Multimedia Publication, Pages (158-161), October 16th - 19th, 2005.
 14. Roman Jarina, Noel Murphy, Noel O'Connor, Sean Marlow: **Speech-Music Discrimination from MPEG-1 Bitstream**, Volume 1, pages (129-132), 2002.
 15. Rudra Pratap: **Getting Started with Matlab: a quick introduction for Scientists and Engineer**; Oxford University Press, USA, 2005.
 16. S.Kotsiantis, D. Kanellopoulos, P.Pintelas: **Multimedia Mining**, Departments of Mathematics, Department of Electrical Engineering & computer Technology, University of Patras, Patras 26500, Greece;
 17. Stuart Cunningham: **Matlab Auditory Demonstration**, Speech and Hearing Departments of Computer Science, University of Sheffield.
 18. Tobias Anderson: *Audio Processing and Transport Multimedia Technologies*, Ericsson Research, Corporate Unit, Lulea, Sweden, March, 2004
 19. Tong Zhang and C.C.Jay Kuo: **Heuristic Approach for Generic Audio data Segmentation and Annotation**; ACM multimedia Transaction, Florida, USA, pages (67-76), 1999.
 20. Toni Heittola: **Automatic Classification of Music Signal**, Master of Science Thesis, Department of Information Technology, Tampere University of Technology, 14th February, 2003.
 21. Vlad M. Trifa: **A Framework for Bird Song Detection, Recognition, and Localization using Acoustic Sensor Networks**, Master's Thesis, February 16th, 2006
 22. <http://cslu.cse.ogi.edu/HLTsurvey/ch1node4.html>
 23. http://www2.cs.uregina.ca/~dbd/cs831/notes/confusion_matrix/confusion_matrix.html
 24. <http://www.twocrows.com/glossary.htm#anchor311516>
 25. www.wikipedia.org
 26. www.weka.net.nz3