DOKUZ EYLÜL UNIVERSITY
GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

A COMPARISON OF DIFFERENT CLASSIFICATION SYSTEMS FOR AUTOMATIC SINGER IDENTIFICATION

by
Emrah KARAMAN

September, 2009
İZMİR

A COMPARISON OF DIFFERENT CLASSIFICATION SYSTEMS FOR AUTOMATIC SINGER IDENTIFICATION

A Thesis Submitted to the
Graduate School of Natural and Applied Sciences of Dokuz Eylül University
In Partial Fulfillment of the Requirements for the Degree of Master of Science
in Electrical and Electronics Engineering Program

by
Emrah KARAMAN

September, 2009
İZMİR

M.Sc THESIS EXAMINATION RESULT FORM

We have read the thesis entitled **“A COMPARISON OF DIFFERENT CLASSIFICATION SYSTEMS FOR AUTOMATIC SINGER IDENTIFICATION”** completed by **EMRAH KARAMAN** under supervision of **ASST. PROF. DR. DAMLA KUNTALP** and we certify that in our opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

....

Asst. Prof. Dr. Damla KUNTALP

Supervisor

....

(Jury Member) (Jury Member)

Prof.Dr. Cahit HELVACI
Director
Graduate School of Natural and Applied Sciences

ACKNOWLEDGEMENTS

I would like to thank my adviser, Asst.Prof.Dr. Damla KUNTALP, for her endless and valuable support in guidance and supervision for this project.

I am very grateful to my family for their understanding and support during my educational background as in whole my life.

Emrah KARAMAN

A COMPARISON OF DIFFERENT CLASSIFICATION SYSTEMS FOR AUTOMATIC SINGER IDENTIFICATION

ABSTRACT

In this project, methods for automatic singer identification problem are investigated, and a singer identification system for 15 singers is implemented. The system consists of two parts. Firstly, the song as the input of the system is segmented into two parts: vocal part, which consists of the singers' voice and instrument sounds, and non-vocal part which consists of only instruments' sounds. Then, the identification step for modeling and classification of singer is applied. Both steps consist of the audio feature extraction and classification methods. In the beginning of feature extraction, preprocessing is applied to data such as down-sampling, normalization, pre-emphasizing, frame blocking and windowing. Energy, spectral flux, zero crossing rate, mel frequency cepstrum coefficients (MFCC) and linear prediction cepstrum coefficients (LPCC) are used for feature extraction. Then, support vector machine (SVM), gaussian mixture model (GMM) and multilayer perceptron (MLP) classifiers are constructed for classification of the singer with using all these extracted features.

Keywords: Singer identification, audio feature extraction, MFCC, LPCC, SVM, GMM, MLP

OTOMATİK ŞARKICI TANIMADA FARKLI SINIFLANDIRMA SİSTEMLERİNİN KARŞILAŞTIRILMASI

ÖZ

Bu projede otomatik şarkıcı tanıma problemi için yöntemler incelenmiş ve 15 şarkıcı için bir otomatik şarkıcı tanıma sistemi oluşturulmuştur. Sistem iki aşamadan oluşmaktadır. Öncelikle sistem girdisi olarak kullanılan şarkı vokal kısım yani şarkıcının sesiyle beraber enstrüman seslerinin olduğu ve vokal olmayan kısım yani sadece enstrüman seslerinin olduğu kısım olarak ikiye ayrılır. Daha sonra şarkıcı tanıma ve sınıflandırma için tanıma aşaması gerçekleştirilir. Her iki aşamada da ses öznitelikleri çıkarma ve sınıflandırma işlemleri uygulanmaktadır. Öznitelik çıkarma işleminden önce şarkıya örnekleme düşürme, normalleştirme, çerçeveleme ve pencereleme gibi ön işlemler uygulanır. Öznitelik olarak enerji, spektral akı, sıfır geçiş oranı, mel frekansı cepstrum katsayıları (MFCC) ve doğrusal öngörülü cepstrum katsayıları (LPCC) kullanılır. Daha sonra bu öznitelikler kullanılarak şarkıcıyı belirlemek için destek vektör makineleri (SVM), çoklu gauss karışım modelleri (GMM) ve çok katmanlı algılayıcılarla (MLP) sınıflandırıcılar oluşturulur.

Anahtar Sözcükler: Şarkıcı tanıma, ses öznitelikleri, MFCC, LPCC, SVM, GMM, MLP

CONTENTS

Page

CHAPTER ONE
INTRODUCTION

1.1 Singer Identification

Singer identification (SID) which is the task of automatically identifying the singer of a song is important for music indexing and retrieval. Rapid progress in digital media and computer technology makes it more popular day by day.

SID is analogous to speaker identification, which aims to determine who is speaking, but it is more complex due to fact that there are significant differences between singing and speaking. First of all, the singing voice is mixed with musical instrumental sounds in a song, which makes it much more complicated to extract features of only the voice. Furthermore, the time-frequency features of a singing voice are quite different from those of a speaking voice. It is therefore important to focus on the vocal part in polyphonic sound mixtures to prevent the negative influences of accompaniment sounds.

SID is an emerging field spurred by the rapid proliferation of popular music on the Internet. In contrast to classification of complete songs based on genre, singer ID can be used to find cameo’s or guest appearances in live concert recordings, to identify the singers in a movie’s musical interludes, to distinguish between an original song and a cover-band, or otherwise to obtain singer identity information where it may be undocumented or difficult to find. Furthermore, SID may also enable companies to rapidly scan suspected websites for piracy – especially bootleg concert recordings, in which the company will typically not have a copy of the original audio data for comparison. (Tsai, Wang, & Rodgers, 2003)

An SID system has three fundamental steps detailed in Chapter II.

1) Segmentation step that separates the vocal and non-vocal segments in a song. A vocal segment can be a mixture of vocals with or without instrumental background.

2) Feature extraction step that obtains the singer’s feature parameters from the vocal segments. The feature parameters are usually presented in the frequency domain to capture the singer’s acoustic characteristics.

3) Classification step that is trained using an individual singer’s voice. During identification, when presented an unknown vocal segment, the classifier identifies the singer.

1.2 Literature Review

In this part, we present a literature review of the work related to singer identification. We begin with a general overview of datasets, features, classifiers and segmentation. Following this, we review works that is directly related to singer identification.

Music is becoming an important part of daily life with the improvement on devices using digital media. Also, rapid progress in computer and internet technology has enabled the circulation of large amounts of music data on the Internet (Nwe, & Li, 2007). Therefore, it is not hard to create a database for singer identification. Most of the people worked on this task have used wav files converted from CD recordings. They were down-sampled from CD sampling rate and bit rate was reduced to exclude the high frequency components beyond the range of normal singing voice. Another advantage of this process was efficient storage area and high processing speed.

There are basically two types of features used on audio processing: one is temporal and the other one is spectral. The temporal features and most of the spectral features are not efficient in singer identification because of the mixed structure in time domain and similarity in frequency domain of singer voice and instruments’ sounds. Therefore, cepstral coefficients, which give information for measuring formant or the smooth spectral envelope, are most common features in singer identification task. The used features are Mel Frequency Cepstrum Coefficients (MFCC), Octave Scale Cepstrum Coefficients (OSCC), Low Frequency Power Coefficients (LFPC), Linear Prediction Coefficients (LPC), Linear Prediction

Cepstrum Coefficients (LPCC), energy, spectral flux, vibrato, zero-crossing rate and other acoustic features.

The singer identification task is a pattern recognition problem. In most of the systems, supervised pattern recognition techniques namely Gaussian Mixture Models (GMM), Support Vector Machines (SVM), Artifical Neural Networks (ANN) and Hidden Markov Models (HMM) were used.

Segmentation is an important step in singer identification. Because the features should be extracted from the singer's voice for an efficient SID system. For this purpose, the instruments' sounds should be eliminated as much as possible. This step was done manually in earlier works. Then, it was done with using some algorithm such as vocal detector to make the system completely automatic.

One of the first and important singer identification systems was done by using ANN and SVM classifiers which were applied to spectral features (Whitman, Flake, & Lawrance, 2001). They did not segment vocal and non-vocal parts. The system performance was not impressive. It achieved a 70% accuracy in a 10-singer database, and 50% accuracy in a 20-singer database. Using the same database, another system was developed in 2002. Its accuracy was improved to be 65% in the 20-singer database case (Berenzweig, Ellis, & Lawrence, 2002). They used a vocal vs. non-vocal detector during pre-processing and only vocal segments were added to training data.

A standard text-independent speaker identification method was applied to singer identification task (Zhang, 2003). Zhang used only vocal segments, extracted MFCCs and modeled each singer with a GMM. He got an 82% accuracy but the vocal segments was collected manually and the database was small. This system was improved by adding non-vocal segments (Tsai, Wang, & Rodgers, 2003). They segmented the vocal and non-vocal parts of each song and modeled each singer with also using these segments by using GMM. The performance was 80%. They neglected the correlation between the background music and the singer.

Another system was developed using HMM (Kim, 2003). In this system, the data set consisted of recordings from 4 conservatory trained classical singers (two sopranos, one tenor, and one bass baritone). Each singer performed a variety of vocal exercises emphasizing the 5 major vowels. The identification system operated by determining which singer's HMM had the highest likelihood for a given vowel, and the singer with the most vowel HMM matched was identified as the source performer. Its accuracy was above 70%.

In another study, the rhythm structure of the song was analyzed using a rhythm tracking method and the song was segmented into beat space time frames. This inter-beat time resolution of the song was used for both feature extraction and training of the classifiers. This enabled to use musically meaningful inter-beat-interval, instead of conventional 20~30 ms frame length as the time resolution for music segmentation (Maddage, Xu, & Wang, 2004). They also used OSCCs which were not used in previous systems. The identification of the singer had an accuracy of over 87% by correlating both vocal and non-vocal GMMs. This system had a high accuracy when it compared with previous systems but it was more computationally complex.

A new method, the computation of a robust estimate of the spectral envelope called the composite transfer function (CTF), was used to identify singer (Bartsch, & Wakefield, 2004). The CTF is derived from the instantaneous amplitude and frequency of the sinusoidal partials which make up the vocal signal. The principal components of the CTFs were used as features for a quadratic classifier to identify singers. Restricting the frequency range of the CTFs and using a test set containing samples extracted from solo performances of Italian arias decreases the accuracy to 70–80%.

Two methods, accompaniment sound reduction and reliable frame selection, were developed to reduce the negative effects of accompaniment sounds on singer identification (Fujihara, Kitahara, Goto, Komatani, Ogata, & Okuno, 2005). They got a 95% accuracy over forty songs by ten singers dataset. Solo voice modeling technique was improved to reduce the effects of accompaniment sounds by leveraging statistical estimation of a piece's musical background (Tsai, & Wang, 2006). The performance was above 80%.

In some studies, some improvements were done on feature extraction and classifier design. MFCCs were divided into two subsets: the low order MFCCs characterizing the vocal tract resonances and the high order MFCCs related to the glottal wave shape (Mesaros, & Astola, 2005). Classification strategies included the discriminant functions, GMM - based maximum likelihood classifier and nearest neighbour classifiers using Kullback-Leibler divergence between the GMMs (Mesaros, Virtanen, & Klapuri, 2007).

Latest important system explored vibrato characteristics of singers present in different sections of a song. Vocal detector and singer classifier were HMM-based classifiers following the bayes decision rule. It had an 84% accuracy.

1.3 Outline

This thesis is presented in six chapters. Chapter 1 presents introduction and literature review. Chapter 2 introduces overview of the system. Chapter 3 gives theoretical background of each step in an SID system. Chapter 4 presents the system which is implemented. Results are represented in Chapter 5 and finally Chapter 6 gives the conclusion and possible future work aspects.

CHAPTER TWO
OVERVIEW OF THE SYSTEM

SID system is a classification procedure consists of two steps: segmentation and singer modeling. Vocal and non-vocal parts of a song are separated from each other in segmentation step. By using either only vocal parts or both vocal and non-vocal parts, singer of the song is identified by using multi-model classifier which consists of a classifier of each singer. The steps given below in detail are applied to each classifier mentioned above.

Steps of an audio classification are similar to classifications on an SID system. First of all, a dataset should be selected to generalize the classification for the problem. Then, this dataset is separated into two groups: one is for training and the other is for testing of the system because of using supervised pattern recognition techniques. Pre-processing is applied to each sample in groups of dataset. Generally, down-sampling to use less storage area and exclude the higher frequency components, normalization to eliminate high peaks, pre-emphasizing to flatten the spectrum, frame blocking to divide the signal into matrix form with appropriate time length for each frame with an overlap and windowing to reduce the signal discontinuity at the end of each block caused by overlapping are done in pre-processing. Then, the features are extracted from each frame in frame matrix and it becomes a feature matrix. Figure 2.1 shows these steps.

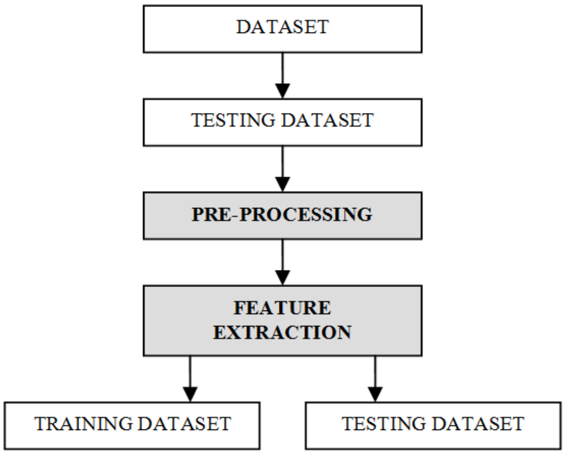

Figure 2.1 Pre-processing and feature extraction

All these steps are applied to all samples in both train and test dataset. Selected classifier is trained with the training dataset after the steps mentioned above. Then, it is tested with testing dataset. An optimum solution, which means classification error is as small as possible, is tried to be found. Figure 2.2 shows the rest of the audio classification.

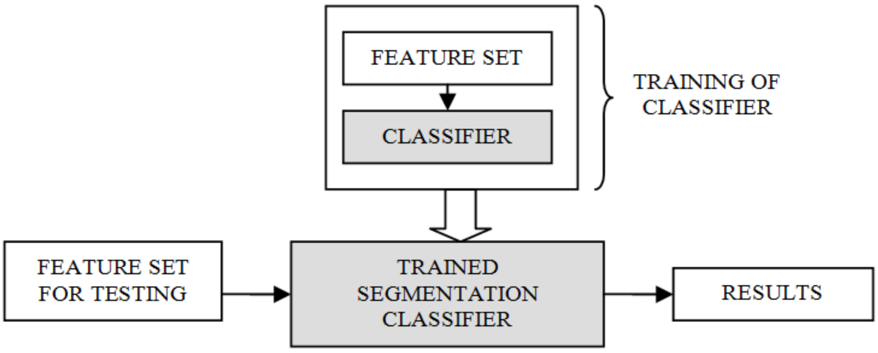

Figure 2.2 Training and testing of classifier for segmentation

Two different datasets are selected for SID system to train and test the each classification models. The dataset will be used for segmentation has to generalize the problem of vocal versus non-vocal separation. Therefore, selecting samples of dataset from different genres independent of singers increase this generalization. This dataset, manually segmented to vocal and non-vocal parts, is divided into two groups for training and testing phases which are demonstrated in Figure 2.3.

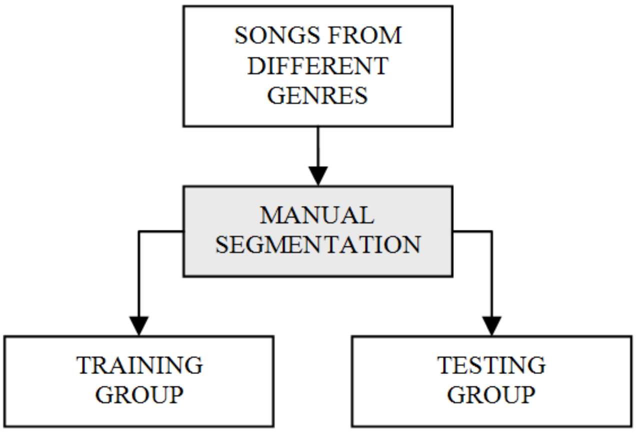

Figure 2.3 Dataset for segmentation

Identification is a multi-classification problem since the aim is identifying the singer of a song among a group of singers. A classifier is modeled for each singer. Therefore, a dataset is compiled for each singer by selecting his/her songs. In contrast to segmentation dataset, it has to generalize the characteristics of the singer voice. Figure 2.4 presents the identification dataset.

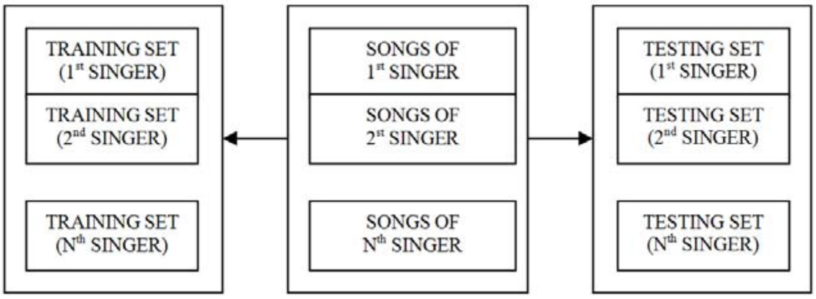

Figure 2.4 Dataset for singer modeling and identification

After selection of datasets, the classification technique is chosen and SID system is constructed according to classification procedure detailed above.

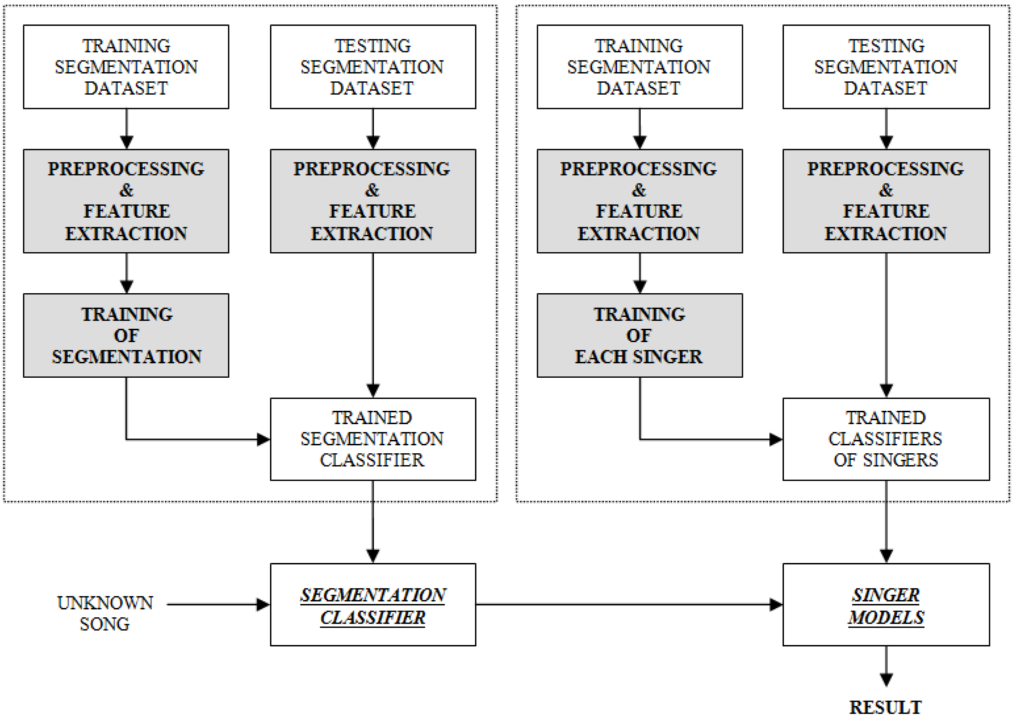
TRAINING SEGMENTATION DATASET
TESTING SEGMENTATION DATASET
PREPROCESSING & FEATURE EXTRACTION
PREPROCESSING & FEATURE EXTRACTION
TRAINING OF SEGMENTATION
TRAINED SEGMENTATION CLASSIFIER
TRAINING SEGMENTATION DATASET
TESTING SEGMENTATION DATASET
PREPROCESSING & FEATURE EXTRACTION
PREPROCESSING & FEATURE EXTRACTION
TRAINING OF EACH SINGER
TRAINED CLASSIFIERS OF SINGERS
UNKNOWN SONG
SEGMENTATION CLASSIFIER
SINGER MODELS
RESULT

Figure 2.5 System design for SID problem

CHAPTER THREE
THEORY

In this chapter theoretical background information is given for the pre-processing, feature extraction and classification methods that are used in SID system presented in previous chapter.

3.1 Preprocessing Methods

Pre-processing is an essential part before feature extraction and as the name implies, preprocessing involves the conditioning of digital music signal prior to extracting the specific features from the music signal. Figure 3.1 displays the steps of pre-processing.

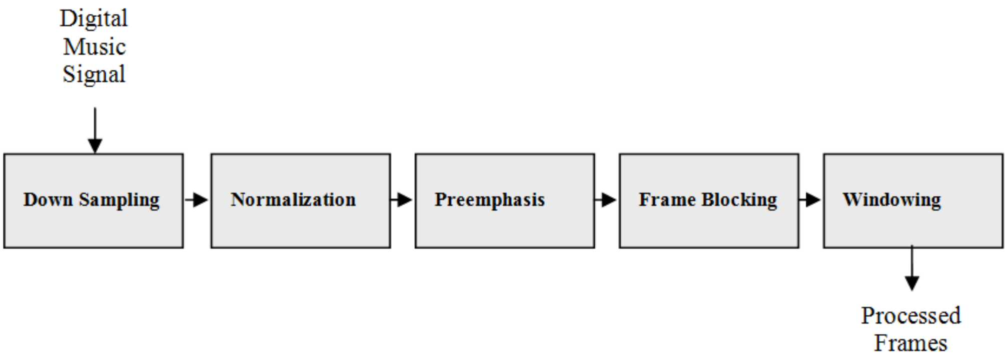

Figure 3.1 Pre-processing steps

3.1.1 Down-Sampling

Down-sampling is the process of reducing the sampling rate of a signal. This is usually done to reduce the size of the data. The down-sampling factor is usually an integer or a rational fraction greater than unity. This factor multiplies the sampling time or, equivalently, divides the sampling rate.

3.1.2 Normalization

Normalization is the process of increasing (or decreasing) the amplitude of an entire audio signal so that the resulting peak amplitude matches a desired target. Typically, normalization increases the amplitude of the audio waveform to the maximum level that does not introduce any new distortion other than that of re-quantization.

3.1.3 Pre-emphasis

The aim of pre-emphasis is to compensate the high-frequency part that was suppressed during the sound production mechanism of humans. The audio signal $X(n)$ is sent to a first order finite-input response high pass filter,

$$Y(n) = X(n) - a * X(n-1) \tag{3.1}$$

where $Y(n)$ is the output signal and the value of a is usually between 0.9 and 1. The z-transform of the filter is

$$\mathrm{H(z)} = 1 - a * \mathrm{z}^{-1} \tag{3.2}$$

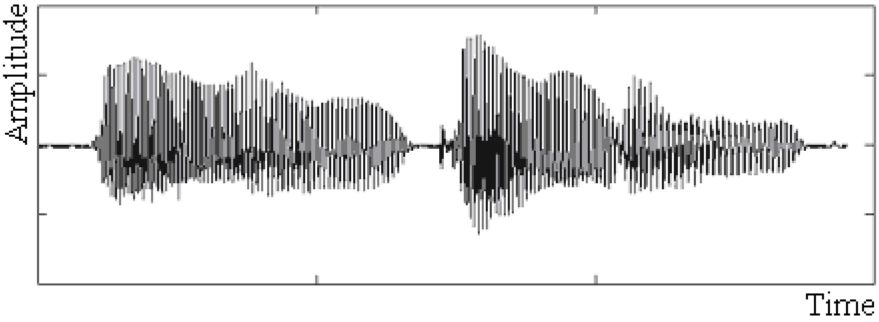

Figure 3.2 A music signal example

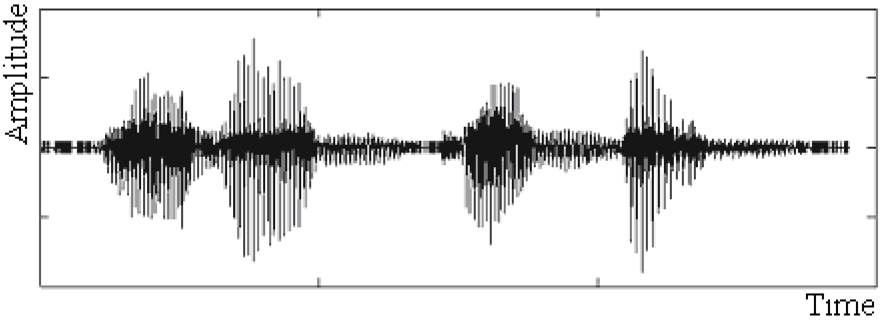

Figure 3.3 Pre-emphasized music signal example

3.1.4 Frame Blocking

The music signal can be considered a quasi-stationary signal. A frame example of music signal is shown in Figure 3.4. When examined over a sufficiently short period of time (between 5 and 100 msec), its characteristics are fairly stationary. Therefore, short-time spectral analysis is the most common way to characterize the music signal. Most of the state-of-the-art systems today use frame duration between 10 and 50 msec. In addition, overlapping is also applied while frame blocking to prevent data loss between frames.

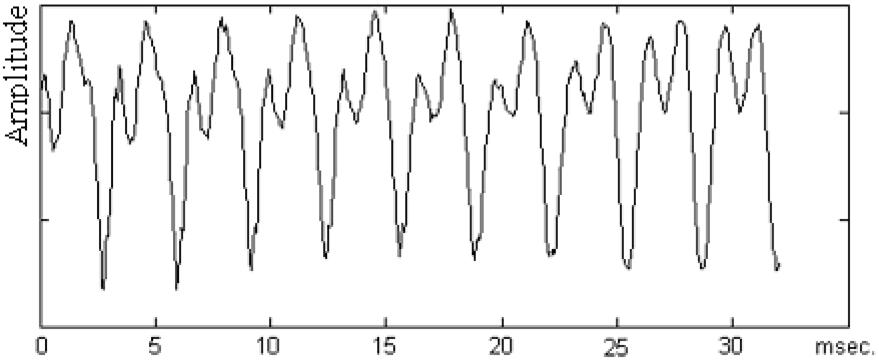

Figure 3.4 A frame example (32msec)

3.1.5 Windowing

The last step in pre-processing is to window each individual frame so as to minimize the signal discontinuities at the beginning and end of each frame. The concept here is to minimize the spectral distortion by using the window to taper the signal to zero at the beginning and end of each frame. If we define the window as $\mathrm{w(n)}$, $0 \le \mathrm{n} \le N-1$, where N is the number of samples in each frame, then the result of windowing the signal is given as:

$$\mathrm{y_1(n) = x_1(n) * w(n)} \tag{3.3}$$

Different window shapes are realized by applying a weighting function. Most typical is the Hamming window with $\alpha_{\mathrm{w}} = 0.54$,

$$w(n) = \frac{\alpha_w - (1-\alpha_w)\cos(2\pi\nu/(N_s - 1))}{\beta_w} \qquad (3.4)$$

3.2 Feature Extraction

It is known that the individual characteristics of audio signals are expressed in their spectral envelopes. In the field of speech recognition studies, various methods have been proposed for calculating feature vectors concerning spectral envelopes. Here, some of them are given in detail, which are commonly used in speech recognition and speaker identification studies.

3.2.1 Energy

It represents the amplitude variation over the time of the audio signal. The start of singing voice is normally reflected as a sudden rise in the energy level of the audio signal. A typical example is illustrated in Figure 3.5 where the short-time energy function of a song is displayed, and the arrow indicates the starting point of the singing voice. Also, in some songs, the appearance of low level local energy minimums after a relatively long period of continuously high energy values may indicate the start of the singing voice.

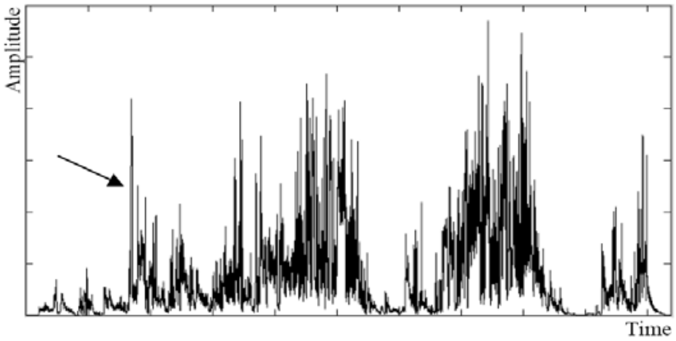

Figure 3.5 Short-time energy function of a song

3.2.2 Zero Crossing Rate

Zero crossing rate is the rate of sign-changes which are from positive to negative or vice versa along a signal. This feature is defined formally as;

$$zcr = \frac{1}{L}\sum_{n=0}^{L-1}\Gamma\left\{X_n X_{n-1} < 0\right\} \tag{3.5}$$

where X the signal of length L and the indicator function $\Gamma\{A\}$ is 1 if its argument A is true and 0 otherwise. While ZCR values of instrumental music are normally within a relatively small range, the singing voice is often indicated by high amplitude ZCR peaks resulted from pronunciations of consonants. An illustration is shown in Figure 3.6, where the short-time average zero-crossing rates of a song are plotted, and the arrow denotes the start of the singing signal.

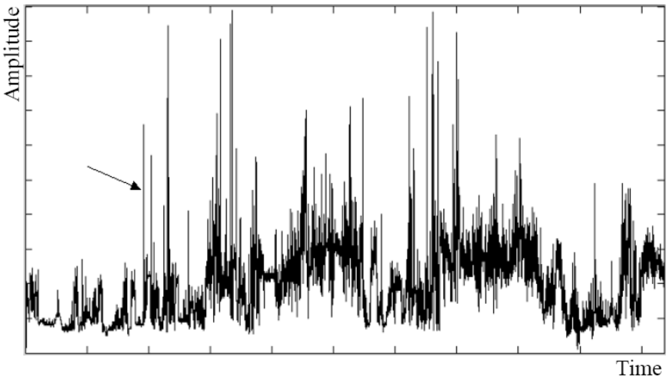

Figure 3.6 Short-time average zero-crossing rates of a song

3.2.3 Spectral Flux

It is defined as the 2-norm of the frame to frame spectral amplitude difference vector:

$$\mathrm{F_n} = \left\| \left|\mathrm{X_n}(\omega)\right| - \left|\mathrm{X_{n+1}}(\omega)\right| \right\| \tag{3.6}$$

where $\left|\mathrm{X_n}(\omega)\right|$ is the magnitude spectrum of the n^{th} frame of the audio signal. The start of singing voice is often indicated by the appearance of high peaks in the spectral flux value, because the voice signal tends to have higher rate of change than instrumental music. An example is shown in Figure 3.7, where the arrow denotes the start of the singing voice in the spectral flux values of a song.

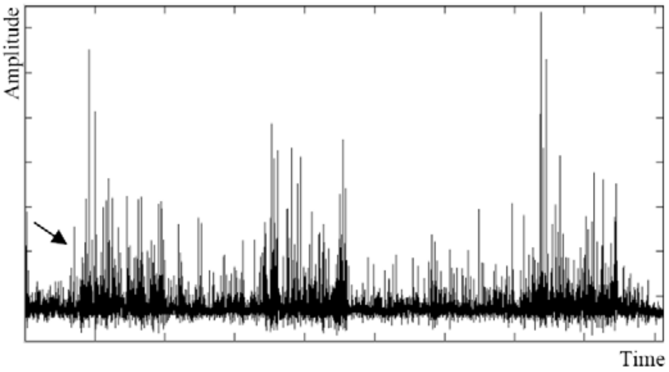

Figure 3.7 Short-time average zero-crossing rates of a song

3.2.4 Cepstral Coefficients

Formants are spectral prominences created by one or more resonances in the sound source. They represent essential information for speech and speaker recognition, and also for musical instrument recognition. A robust feature for measuring formant information, or the smooth spectral envelope, is cepstral coefficients. The cepstrum of a signal *y(n)* is defined as

$$c(n) = F^{-1}\{\log\left|F\{y(n)\}\right|\} \tag{3.7}$$

where F stands for the Discrete Fourier transform (DFT). Calculating cepstral coefficients from the above equation is not very efficient, since two Fast Fourier transforms (FFT) are needed. The coefficients can be more efficiently calculated from a mel-frequency filterbank or from linear prediction coefficients.

Another reason for not using the above equation is the utilization of psychoacoustic frequency scales. DFT uses linear frequency resolution, so we must use some kind of warping transform to convert the linear frequency scale into a perceptual scale.

3.2.5 Mel Frequency Cepstral Coefficients

Mel-Frequency Cepstral Coefficients (MFCC) has become one of the most popular techniques for the front-end feature-extraction in automatic speech recognition or speaker identification systems. We will use here the conventional FFT-based method utilizing a mel-scaling filterbank. Figure 3.8 shows a block diagram of the MFCC feature extractor.

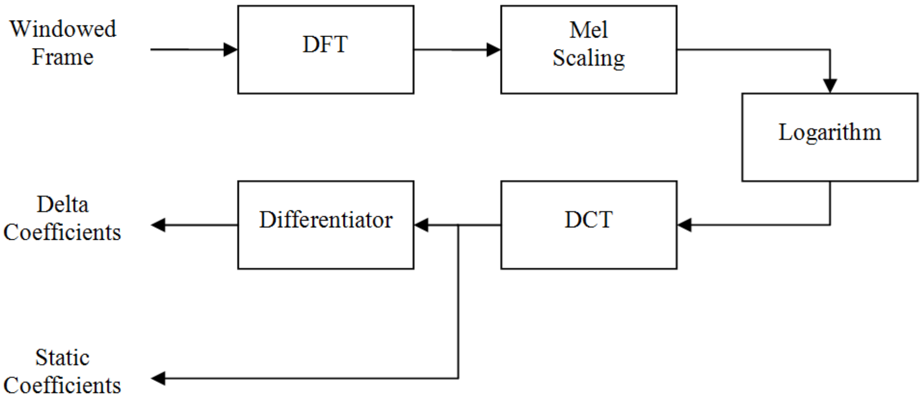

Figure 3.8 Mel frequency cepstral coefficients

Next, a filterbank consisting of triangular filters spaced uniformly across the mel-frequency scale and their heights scaled to unity is simulated. The mel-scale is given by

$$Mel(f) = 2595\log_{10}(1+\frac{f}{700}) \tag{3.8}$$

where *f* is the linear frequency value. To implement this filterbank, a window of audio data is transformed using the DFT, and its magnitude is taken. By multiplying the magnitude spectrum with each triangular filter and summing the values at each channel, a spectral magnitude value for each channel is obtained.

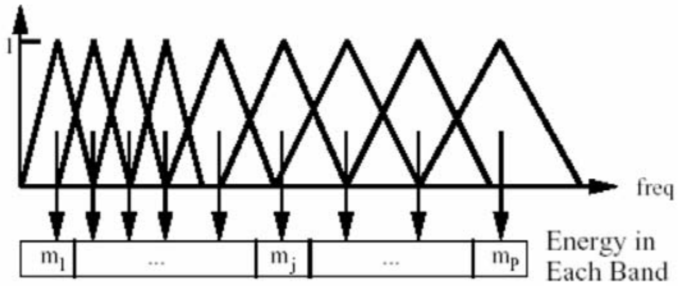

Figure 3.9 Mel scale fitler bank

The dynamic range of the spectrum is compressed by taking a logarithm of the magnitude at each filterbank channel. Finally, cepstral coefficients are computed by applying a discrete cosine transform (DCT) to the log filterbank magnitudes m_j as follows:

$$c_{mel}(i) = \sum_{j=1}^{N} m_j \cos\left(\frac{\pi i}{N}\left(j - \frac{1}{2}\right)\right) \tag{3.9}$$

DCT decorrelates the cepstral coefficients, thereby making it possible to use diagonal covariance matrices in the statistical modeling of the feature observations.

3.2.6 Linear Prediction Cepstrum Coefficients

Linear prediction analysis is a method for estimating the transfer function of vocal tract, assuming that input audio signal contains only human voice. Linear prediction coefficients (LPC) predict the current sample of the audio signal from a linear combination of past samples. The LPC algorithm is an n^{th} order predictor which attempts to predict the value of any point in a time-varying linear system based on the values of previous *n* samples. The representation of the vocal tract transfer function, $H(\mathrm{z})$, can be given by the following equation:

$$H(\mathrm{z}) = \frac{G}{1 - \sum_{i=1}^{\mathrm{P}} a(i)\mathrm{z}^{-i}} \tag{3.10}$$

The values $a(i)$ are called the *prediction coefficients*, while G represents the amplitude, or gain, associated with the vocal tract excitation. The notation z^{-1} in the domain of z-transform represents a system function and corresponds to a unit delay in the time domain. For discrete-time signals, the z-transforms can be considered a generalization of the Fourier transform. The poles of the transfer function in Equation (3.10) are determined from the roots of the polynomial in the denominator. The LPC can only derive the resonant frequencies, or the formants, but not the zeros. The LPC does not adequately estimate signals that have no poles, such as some unvoiced speech noise. The non-linear signal components adversely affect the LPC estimates.

Cepstrum can be calculated in two ways, one is by using simple recursion and the other is with the Fourier transform.

Using the Linear Prediction Coefficients: The LPC-derived cepstrum coefficients are defined as follows, where c_i is the i^{th} cepstrum coefficient and a_k are the prediction coefficients :

$$c_1 = a_1 \tag{3.11}$$

$$c_i = a_i + \sum_{k=1}^{i-1}((1-\frac{k}{i})a_k c_{i-k}) \tag{3.12}$$

Unlike LPC coefficients, cepstrum coefficients are independent and the distance between cepstrum coefficients vectors can be calculated with a Euclidean-type distance measure.

Using the Fourier Method: Music signal $x(n)$ can be expressed as the convolution of glottal pulses $g(n)$ and vocal tract impulse response $v(n)$. In other words,

$$x(n) = g(n) * v(n) \tag{3.13}$$

Letting the logarithmic operation for the discrete Fourier transformation be D ,

$$\mathrm{D}\{x(n)\} = \mathrm{D}\{g(n) * v(n)\} = \mathrm{D}\{g(n)\} + \mathrm{D}\{v(n)\} \tag{3.14}$$

The inverse discrete Fourier transform for $\mathrm{D}\{x(n)\}$ is called a cepstrum. In other words,

$$c(n) = \frac{1}{2\pi}\int_0^{2\pi} \log|\mathrm{X}(\omega)|\mathrm{e}^{\mathrm{j}n\omega}\mathrm{d}\omega \tag{3.15}$$

$$\mathrm{X}(\omega) = |\mathrm{X(z)}|_{\mathrm{z}=\mathrm{e}^{\mathrm{j}\omega\mathrm{t}}} \tag{3.16}$$

The cepstrum for $x(n)$ turns out to be the sum of the cepstrum for $g(n)$ and the cepstrum for $v(n)$. The independent variable of the cepstrum has a time dimension (frequency). In the case of a voiced sound, $\mathrm{D}\{g(n)\}$ appears as a component in the neighborhood of *1/F₀* (*F₀* is the fundamental frequency) on the time axis, and $\mathrm{D}\{v(n)\}$ as a component of the short time domain. Thus, a window is opened in the cepstrum and the short time range components are extracted (this is accomplished by removing $g(n)$), and if a discrete Fourier transformation is performed in this, the spectral envelope is obtained.

3.3 Classification

Classification is the main part of a SID system as it is mentioned in Chapter II. Supervised learning is used in both segmentation and singer modeling steps. Supervised learning is a machine learning technique for deducing a function from training data. The training data consist of pairs of input objects, and desired outputs. The output of the function can be a continuous value, or can predict a class label of the input object. The task of the supervised learner is to predict the value of the function for any valid input object after having seen a number of training examples.

3.3.1 Statistical Classification

Statistical classification is a procedure in which individual items are placed into groups based on quantitative information on one or more characteristics inherent in the items referred to as traits, variables or characters and based on a training set of previously labeled items. The statistical classifiers are commonly used for music information retrieval problems assume that the data follows a certain distribution, and try to estimate the parameters of the class distributions from the training observations. Knowing the probability density function of the assumed distribution, the likelihood of each class distribution of generating the test observation can then be calculated. Also some other classification methods give applicable results for SID systems.

3.3.1.1 Gaussian Mixture Model

Gaussian mixture model (GMM) classifiers have superior performance for most music information retrieval application. GMM classifiers combine some of the benefits of both the kNN and quadratic classifiers. Like quadratic classifiers, they employ a trainable model that does not require all of the training data to make a classification. However, like kNN classifiers, GMM classifiers with sufficiently high order can approximate any distribution with arbitrary accuracy.

A GMM is a model of the probability density function for a given set of data. The model has the form of a sum of individual Gaussian components, each possessing its own mean and covariance. That is, the probability density function, $f(x)$, as

$$f(x)=\sum_{j=1}^{J}\frac{P(j)}{(2\pi)^{p/2}\,|\Sigma_j|^{1/2}}e^{-\frac{1}{2}(x-\mu_j)^T\Sigma_j^{-1}(x-\mu_j)} \tag{3.17}$$

$$f(x)=\sum_{j=1}^{J}P(j)P(x\mid j) \tag{3.18}$$

where p equal to the dimension of feature space, μ_j and Σ_j are the mean and covariance, $P(x\mid j)$ is the density function of the j^{th} gaussian component and $P(j)$ is the Gaussian mixture probability of the j^{th} component chosen such that

$$\sum_{j=1}^{J}P(j)=1 \tag{3.19}$$

To make use of a Gaussian mixture model, it is necessary to train the model by determining a set of model parameters $\{P(j),\mu_j,\Sigma_j\}$. A well known method that yields locally optimal parameters for a given data set is the *Expectation Maximization (EM) algorithm.* The EM algorithm operates by iterating between the estimation of the mixture probabilities $P(j)$ and the estimation of the individual Gaussian parameters, μ_j and $z^{(i)}$. When using the EM algorithm to train a GMM, the model parameters are set to some initial, and often random, starting stage. Then, given a set of N instances $\{x_n\}_{n=1}^{N}$ and estimates of the Gaussian parameters, the mixture probabilities are estimated as

$$P^{(i)}(j)=\frac{\Sigma_{n=1}^{N}P^{(i-1)}(x_n\mid j)}{\Sigma_{j=1}^{J}\Sigma_{n=1}^{N}P^{(i-1)}(x_n\mid j)} \tag{3.20}$$

Where the notation $z^{(i)}$ indicates the estimate of z computed at iteration *i.* With an estimate of the mixture parameters, the component parameters are estimated by first computing the estimated posterior probabilities of each instance as

$$P^{(i)}(j \mid x_n) = \frac{P^{(i)}(j) P^{(i-1)}(x_n \mid j)}{\Sigma_{k=1}^{J} P^{(i)}(k) P^{(i-1)}(x_n \mid k)} \tag{3.21}$$

Then, these posterior probabilities are used to estimate the mean of each component, as

$$\mu_j^{(i)} = \frac{\Sigma_{n=1}^{N} P^{(i)}(j \mid x_n) x_n}{\Sigma_{n=1}^{N} P^{(i)}(j \mid x_n)} \tag{3.22}$$

The covariance of the Gaussian components can be estimated using a number of methods, depending upon the desired degrees of freedom. If full covariance matrices are desired, they are estimated as

$$\Sigma_j^{(i)} = \frac{\Sigma_{n=1}^{N} P^{(i)}(j \mid x_n)(x_n - \mu_j^{(i)})(x_n - \mu_j^{(i)})^T}{\Sigma_{n=1}^{N} P^{(i)}(j \mid x_n)} \tag{3.23}$$

Full covariance matrices, however, can require a large amount of data to train effectively. One commonly used alternative is the use of diagonal covariance matrices, in which case the covariance matrix can be taken as the diagonal of $\Sigma_j^{(i)}$. Once the model parameters have been estimated, the EM algorithm returns to equation (3.20) and continues to iterate between the estimation of model parameters and the posterior probabilities.

The implementation of the EM algorithm for training a GMM requires several supplemental decisions. As we noted, initial parameter estimates must be determined in some way. Additionally, the case of singular covariance matrices must be handled, usually by imposing a lower limit on the variance for each dimension. Occasionally, components will have mixture probabilities that become very small; these components can either be eliminated from the mixture or introduced at a new random location. Finally, a stopping criterion is needed. Since a Standard iteration of the EM algorithm is guaranteed not to decrease the likelihood of the data given the learned model, thresholds on either the likelihood or its rate of change are common choices.

To use a set of trained GMMs as a classifier, one chooses either a maximum likelihood or maximum a *posteriori* rule, as in the case of the quadratic classifier. Then, the class with the highest likelihood of producing an instance (or a set of instances) is selected as the class label. Convergence of the EM algorithm for training GMMs typically occurs within relatively few iterations, often on the order of 10-20. However, the training of GMM classifiers using the EM algorithm can be computationally expensive, generally requires more data than a quadratic classifier, and is sensitive to initial conditions.

3.3.2 Linear Classification

A linear classifier groups items that have similar feature values into groups by making a classification decision based on the value of the linear combination of the features. If the input feature vector to the classifier is a real vector $\boldsymbol{x}$, then the output score is

$$y = f(\boldsymbol{w} \cdot \boldsymbol{x}) = f\left(\sum_i w_i x_i \right) \tag{3.24}$$

where $\boldsymbol{w}$ is a real vector of weights and f is a function that converts the dot product of the two vectors into the desired output. The weight vector $\boldsymbol{w}$ is learned from a set of labeled training samples. Often f is a simple function that maps all values above a certain threshold to the first class and all other values to the second class. A more complex f might give the probability that an item belongs to a certain class.

For a two-class classification problem, one can visualize the operation of a linear classifier as splitting a high-dimensional input space with a hyperplane: all points on one side of the hyperplane are classified as "yes", while the others are classified as "no".

A linear classifier is often used in situations where the speed of classification is an issue, since it is often the fastest classifier, especially when $\boldsymbol{x}$ is sparse. However, decision trees can be faster. Also, linear classifiers often work very well when the number of dimensions in $\boldsymbol{x}$ is large, as in document classification, where each element in $\boldsymbol{x}$ is typically the number of occurrences of a word in a document.

3.3.2.1 Support Vector Machine

Support vector machines (SVMs) are a set of related supervised learning methods used for classification and regression. Viewing input data as two sets of vectors in an n-dimensional space, an SVM constructs a separating hyperplane in that space, one which maximizes the margin between the two data sets. To calculate the margin, two parallel hyperplanes are constructed, one on each side of the separating hyperplane, which are "pushed up against" the two data sets. Intuitively, a good separation is achieved by the hyperplane that has the largest distance to the neighboring datapoints of both classes, since in general the larger the margin the lower the generalization error of the classifier.

The training data, a set of points of the form, is given as $D = \{(\mathbf{x}_i, c_i) \mid \mathbf{x}_i \in \mathbb{R}^p, c_i \in \{-1,1\}\}_{i=1}^{n}$ where the c_i is either 1 or -1, indicating the class to which the point $\mathbf{x}_i$ belongs. Each $\mathbf{x}_i$ is a *p*-dimensional real vector. The goal is to find the maximum-margin hyperplane which divides the points having $c_i = 1$ from those having $c_i = -1$. Any hyperplane can be written as the set of points $\mathbf{x}$ satifying $\mathbf{w} \cdot \mathbf{x} - b = 0$ where '$\cdot$' denotes the dot product. The vector $\mathbf{w}$ is a normal vector perpendicular to the hyperplane. The parameter, $\frac{b}{\|\mathbf{w}\|}$, determines the offset of the hyperplane from the origin along the normal vector $\mathbf{w}$. $\mathbf{w}$ and b are chosen to maximize the margin or distance between the parallel hyperplanes that are as far apart as possible while still separating the data. These hyerplanes can be described by the equations $\mathbf{w} \cdot \mathbf{x} - b = 1$ and $\mathbf{w} \cdot \mathbf{x} - b = -1$.

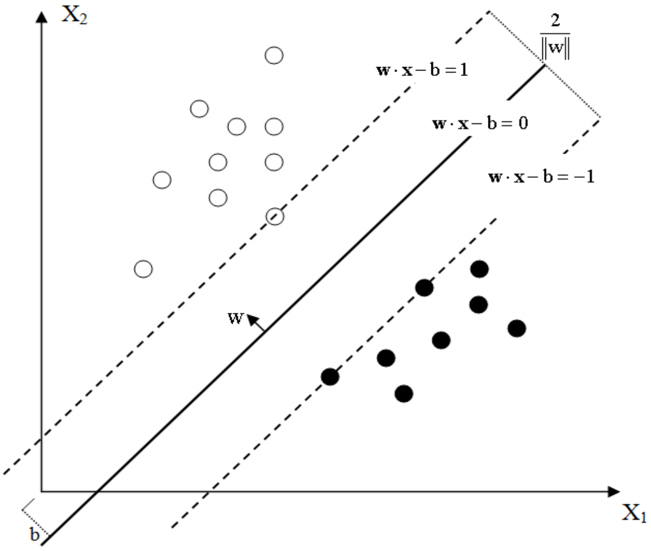

Figure 3.10 Support vectors

3.3.3 Artifical Neural Networks

An artificial neural network (ANN), usually called neural network (NN), is a mathematical model that tries to simulate the structure and/or functional aspects of biological neural networks. It consists of an interconnected group of artificial neurons and processes information using a connectionist approach to computation. In most cases an ANN is an adaptive system that changes its structure based on external or internal information that flows through the network during the learning phase. Neural networks are non-linear statistical data modeling tools. They can be used to model complex relationships between inputs and outputs or to find patterns in data.

3.3.3.1 Perceptron

The perceptron is a type of artificial neural network. It can be seen as the simplest kind of feedforward neural network. The perceptron is a binary classifier that maps its input x to an output value $f(x)$ across the matrix.

$$f(x) = \begin{cases} 1 & \text{if } \; w \cdot x + \mathrm{b} > \mathrm{b} \\ 0 & \text{else} \end{cases} \tag{3.25}$$

where w is a vector of real-valued weights and $w \cdot x$ is the dot product. b is the 'bias', a constant term that does not depend on any input value.

The value of $f(x)$ (0 or 1) is used to classify x as either a positive or a negative instance, in the case of a binary classification problem. The bias can be thought of as offsetting the activation function, or giving the output neuron a "base" level of activity. If b is negative, then the weighted combination of inputs must produce a positive value greater than $-b$ in order to push the classifier neuron over the 0 threshold. Spatially, the bias alters the position (though not the orientation) of the decision boundary.

Since the inputs are fed directly to the output unit via the weighted connections, the perceptron can be considered the simplest kind of feed-forward neural network.

The learning algorithm is the same across all neurons, therefore everything that follows is applied to a single neuron in isolation. Learning is modeled as the weight vector being updated for multiple iterations over all training examples. Let $D_m = \{(x_1, y_1), ..., (x_m, y_m)\}$ denote a training set of m training examples, where x_i is the input vector to the perceptron and y_i is the desired output value of the perceptron for that input vector. Each iteration the weight vector is updated as follows:
For each (x,y) pair in $D_m = \{(x_1, y_1), ..., (x_m, y_m)\}$,

$$w(j) := w(j) + \alpha(y - f(x))x(j) \text{ for } j = 1, \ldots, n \tag{3.26}$$

where $x(j)$ is the j-th item in the n-dimensional input vector, $w(j)$ is the j-th item in the weight vector, $f(x)$ is the output from the neuorun when presented with input x, and α is a constant where $0 < \alpha \leq 1$.

Note that this means that a change in the weight vector will only take place for a given training example (x, y) if its output $f(x)$ is different from the desired output y. The initialization of w is usually performed simply by setting $w(j) := 0$ for all elements $w(j)$.

3.3.3.2 Multi Layer Perceptron

A multilayer perceptron (MLP) is a feedforward artificial neural network model that maps sets of input data onto a set of appropriate output. It is a modification of the standard linear perceptron in that uses three or more layers of neurons with nonlinear activation functions, and is more powerfull than the perceptron in that it can distinguish data that is not linearly separable, or separable by a hyperplane.

If a multilayer perceptron consists neurons uses a nonlinear activation function which was developed to model the frequency of action potentials of biological neurons in the brain. This function is modeled in several ways, but must always be normalizable and differentiable.

The two main activation functions used in current applications are both sigmoids, and are described by $\varphi(v_i) = \tanh(v_i)$ and $\varphi(v_i) = (1 + \mathrm{e}^{v_i})^{-1}$ in which the former function is a hyperbolic tangent which ranges from -1 to 1, and the latter is equivalent in shape but ranges from 0 to 1. Here y_i is the output of the ith node (neuron) and v_i is the weighted sum of the input synapses.

The MLP consists of an input and an output layer with one or more *hidden layers* of nonlinearly-activating nodes. Each node in one layer connects with a certain weight w_{ij} to every other node in the following layer. Figure x.x shows basic MLP.

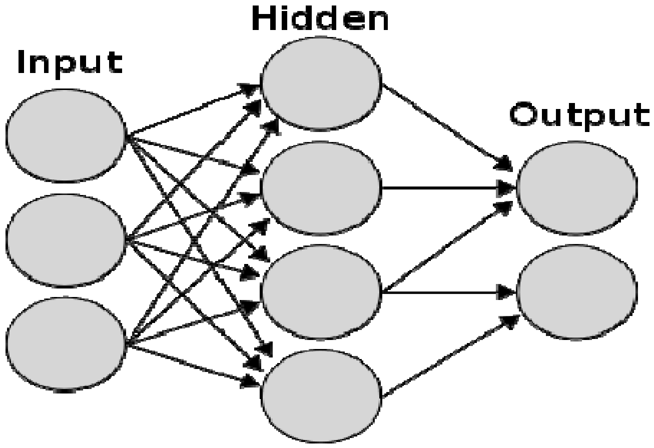

Figure 3.11 A neural network is an interconnected group of nodes.

Learning occurs in the perceptron by changing connection weights (or synaptic weights) after each piece of data is processed, based on the amount of error in the output compared to the expected result. This is carried out through backpropagation, a generalization of the least mean squares algorithm in the linear perceptron. We represent the error in output node j in the n^{th} data point by

$$e_j(n) = d_j(n) - y_j(n) \tag{3.27}$$

where d is the target value and y is the value produced by the perceptron. We then make corrections to the weights of the nodes based on those corrections which minimize the energy of error in the entire output, given by

$$\varepsilon(n) = \frac{1}{2}\sum_j e_j^2(n) \tag{3.28}$$

By the theory of differentials, we find our change in each weight to be

$$\Delta w_{ji}(n) = -\eta \frac{\partial \varepsilon(n)}{\partial v_j(n)} y_i(n) \tag{3.29}$$

where y_i is the output of the previous neuron and η is the *learning rate*, which is carefully selected to ensure that the weights converge to a response that is neither too specific nor too general. In programming applications, typically ranges from 0.2 to 0.8. The derivative to be calculated depends on the input synapse sum v_j, which itself varies. It is easy to prove that for an output node this derivative can be simplified to

$$-\frac{\partial \varepsilon(n)}{\partial v_j(n)} = e_j(n)\varphi^{'}(v_j(n)) \tag{3.30}$$

where $\varphi^{'}$ is the derivative of the activation function described above, which itself does not vary. The analysis is more difficult for the change in weights to a hidden node, but it can be shown that the relevant derivative is

$$-\frac{\partial \varepsilon(n)}{\partial v_j(n)} = \varphi^{'}(v_j(n))\sum_k -\frac{\partial \varepsilon(n)}{\partial v_k(n)} w_{kj}(n) \tag{3.31}$$

Note that this depends on the change in weights of the k^{th} nodes, which represent the output layer. So to change the hidden layer weights, we must first change the output layer weights according to the derivative of the activation function, and so this algorithm represents a *backpropagation of the activation function*.

CHAPTER FOUR
SYSTEM DESIGN AND IMPLEMENTATION

In this chapter, detailed information is given for the SID system studied in this work. Firstly, information for the data is introduced. Then, the vocal characterization and segmentation process are explained. Lastly, singer modeling and identification process is explained.

4.1 Dataset

We used two datasets; one is for segmentation which was collected by Pedro Allegro at INESC Porto (Allegro, 2008) and the other is for singer modeling and identification which is called artist20 set (Ellis, 2007). The segmentation dataset consists of two groups: 24 vocal songs and 24 non-vocal songs. Each song, is sampled at 44.1 kHz, was recorded from CD recordings. The music files contain several different instruments since a lot of musical styles are represented such as jazz, classic, pop, rock, electronic, among others. We take 5 seconds of each song and they are down sampled to 16 kHz to have same sampling rate with the identification dataset. Identification dataset, `artist20`, is a dataset of six albums by each of 20 artists, making a total of 1,413 tracks. Each song, mp3 file, is sampled 16 kHz, and was recorded from CD recordings. Randomly 15 singers and their 30 songs, 30 seconds length, from artist20 are selected for this project. All the songs are converted to **wav** file with same sampling rate. Then, the dataset is divided into three groups for training and testing of singer model, and testing the SID system. The groups for training and testing of singer model are manually segmented to get higher performance.

Finally, the datasets selected for the system are divided into five groups of songs. Table 4.1 represents them and how many songs there are in each group. Identification dataset is divided in three groups. Two of them are manually segmented for training and testing of singer models. Third one is used for final SID system.

Table 4.1 Song groups of datasets used for this project

Groups		Vocal	Non-vocal	Unsegmented
Segmentation	Training	12 songs	12 songs	-
	Testing	12 songs	12 songs	-
Singer Models	Training	160 songs	-	-
	Testing	160 songs	-	-
Overall System	Testing	-	-	160 songs

4.2 Vocal vs. Non-vocal Segmentation

Using only vocal segments for an SID system has achieved 15% improvement, compared to the baseline of the system trained on the unsegmented music signals (Feng, Nielson, and Hansen, 2008). This shows us vocal/non-vocal segmentation has a great influence on the system performance but segmentation is another problem for music information retrieval. Therefore, the performance of segmentation effects the overall performance in great deal. We use three types of classifier: GMM, SVM and MLP. Firstly, we propose to construct a statistical classifier with parametric models trained using accompanied singing voices rather than normal speech. Then, we try to solve this problem by using linear classifier. Lastly, we work on neural networks for segmentation if it has advantage over linear or statistical classifier or not. Segmentation is based on the observation that there is a significant difference in spectral distribution between vocal and instrumental sound. Figure 4.3 shows the spectrogram of a music example.

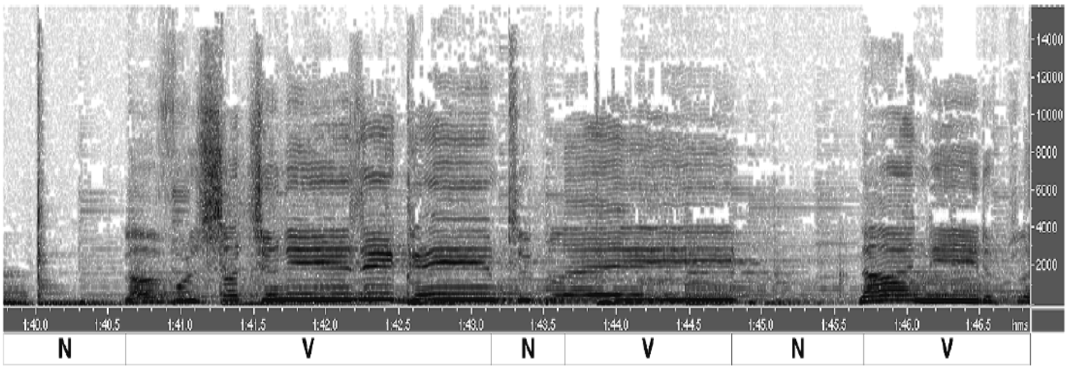

Figure 4.1 Spectrogram of a music example which is “Yesterday” by The Beatles.

Due to the rapid vibration of the vocal folds, the singing voice is nearly always harmonic, and exhibits relatively large amounts of energy at integer multiples of the fundamental frequency in the low or middle frequency regions of the spectrogram (Tsai and Wang, 2003). Compared to the singing voices, the non-vocal parts spread their energy more widely and have less salient harmonics.

As shown in Figure 4.2, the vocal / non-vocal classifier consists of a front-end signal processor that converts music signal into frame-based feature vectors, and a back-end statistical processor that performs modeling, matching and decision making. We investigated three different feature groups: MFCCs, LPCCs, and energy, zero-crossing rate and spectral flux. Features are typically computed using fixed length sliding window, also called a frame, with an overlap. This approach has been used predominantly in speech signal processing, particularly in speech recognition.

The back-end statistical processor operates in two phases: training, and testing. During training of statistical classifier, manually segmented databases are used to form two separate GMM: a vocal GMM, and a non-vocal GMM. Each model consists of several mixture weights, mean vectors and diagonal covariance matrices. The use of GMMs is motivated by the wish to model the spectral distribution of various broad acoustic classes by a combination of Gaussian components. These broad acoustic classes reflect some general vocal and instrumental configurations. It has been shown that GMMs have a strong ability to provide smooth approximations to arbitrarily-shaped densities of a spectrum over a long time span (Reynolds and Rose, 1995).

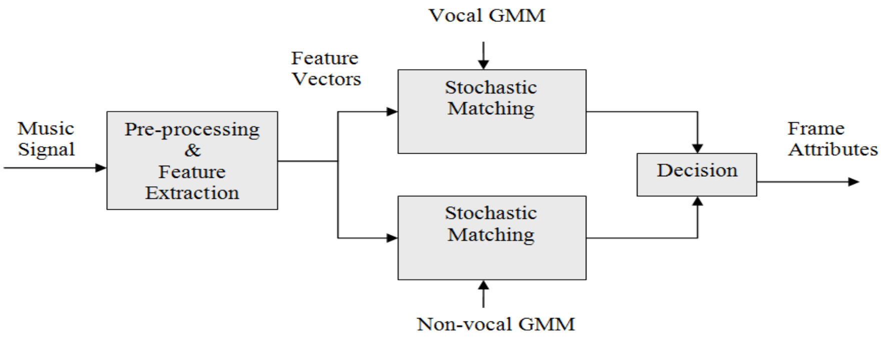

Figure 4.2 Segmentation by GMM

In the testing phase, the classifier takes as input the T_x-frame feature vectors $\mathrm{X} = \{\mathrm{x}_1, \mathrm{x}_2, ..., \mathrm{x}_{\mathrm{T_x}}\}$ extracted from an unknown sample from unsegmented database, and produces as output the frame log-likelihoods $\log p(x_t \mid \lambda_V)$ and $\log p(x_t \mid \lambda_N)$, $1 \le t \le T_x$ for the vocal and non-vocal GMM, respectively. The attribute of each frame is then hypothesized according to a decision rule made on the frame log-likelihoods. Depending on the choice of analysis interval, there are many variations and combinations in decision- making. In this study, we compare only a frame-based decision. For a frame-based decision, the recognizer may trivially hypothesize whether the x_t frame is vocal or not by using

$$\log p(x_t \mid \lambda_V) \underset{non-vocal}{\overset{vocal}{\stackrel{>}{\le}}} \log p(x_t \mid \lambda_N) \tag{4.1}$$

Then we conclude by applying the segmented frame-based song to a moving average filter to make a better decision on segmentation. This filter checks the frame with previous one and next one. Figure 4.3 shows it in detail.

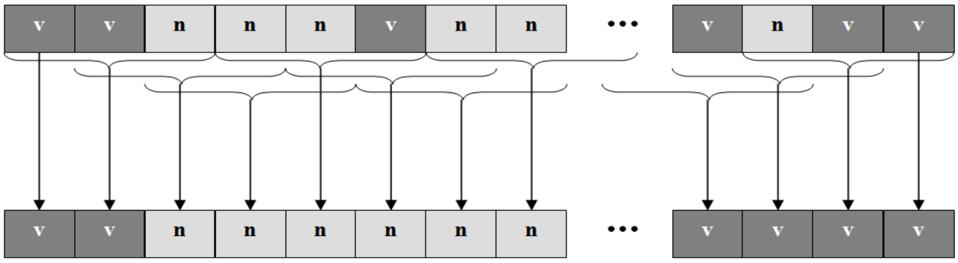

Figure 4.3 Moving average filter application

Finally, segmented song is constructed by choosing the vocal segments according to vocal decided frames numbers.

For other two types of classifier, same feature sets and similar methods are used to solve the segmentation problem. The only difference is in the decision stage. It is easier for SVM classifier because it is already classified into two groups in SVM. We use only vocal classified frames to construct the only-vocal song. Figure 4.4 represents the basic SVM classifier for segmentation.

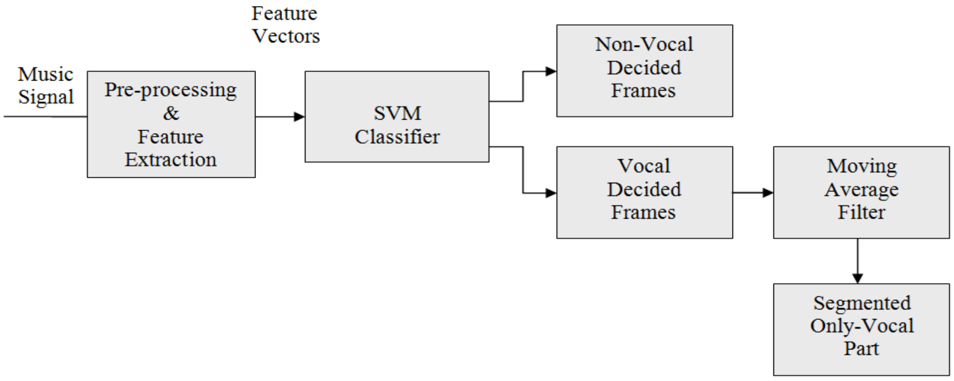

Figure 4.4 Segmentation by SVM

There were many ways to make decision for MLP depending on the activation functions and number of output neurons. We use two methods: one is selecting an output neuron and a threshold to decide the frame belongs to which class and the other is selecting two output neurons with hard-limit transfer function.

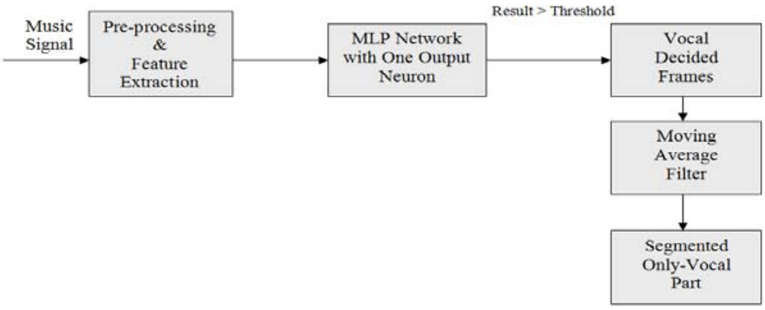

Figure 4.5 Segmentation by MLP with one output neuron

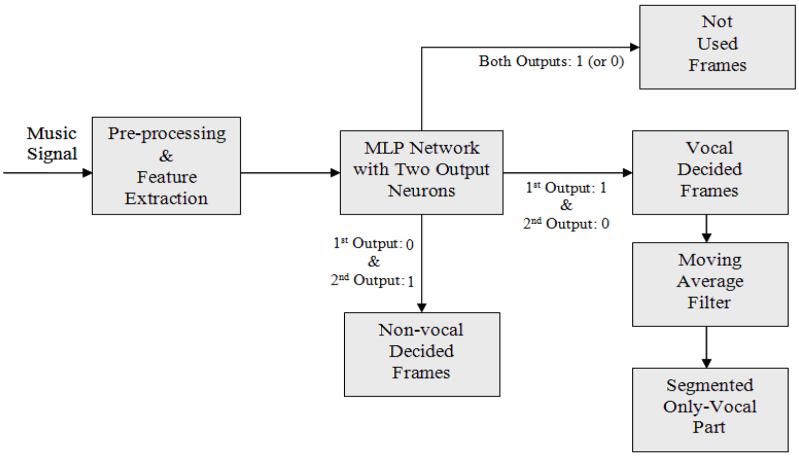

Figure 4.6 Segmentation by MLP with two output neurons

For the first method, *tansig* function is used for the transfer function which varies between -1 and 1. Then, we choose a threshold value. If the output is greater than this value, it assigns the output to 1 otherwise -1. The frames assigned as 1 decided vocal frames. For the other method, if the first output neuron is 1 and the other is 0, it is decided as vocal otherwise non-vocal. If both are 0 or 1 for a frame, it is decided neither vocal nor non-vocal and it is not used.

All the experimental results of segmentation are given in next chapter depending on feature sets and classifier.

4.3 Singer Modeling and Identification

Three classifiers (GMM, SVM and MLP) and two feature (MFCC and LPCC) groups are used for singer modeling.

The basic strategy applied here is to adapt the methods developed in the speaker identification realm to SID. Our baseline system consists of a front-end signal processor that converts music recordings from their digital waveform representations into streams of spectrum-based feature vectors, followed by a backend statistical, linear or other classifier individually that performs modeling and matching. It again operates in two phases, training and testing. During training, the selected segmented songs from dataset are used.

For statistical method, the vocal segments pertaining to each of the singers are used to form a GMM. Under the GMM classifier framework, a set of P singers is represented by voice models $\lambda_{s,1}, \lambda_{s,2}$, … , $\lambda_{s,\mathrm{P}}$ using feature vectors extracted from training data. Parameters of the GMMs are initialized via k-means clustering and iteratively adjusted via expectation -maximization (EM). In the testing phase, the system takes as input the T-length feature vectors $\mathrm{X} = \{x_1, x_2, ..., x_T\}$ extracted from an unknown segmented test recording, and produces as outputs the frame log-likelihoods for all the singer models. Log-likelihoods associated with each singer GMM are accumulated for all the remaining vocal segments. According to the maximum likelihood decision rule, the identifier should decide in favor of a singer S^* satisfying

$$S^* = \underset{1<i\leq \mathrm{P}}{\arg\max} \left\{ \sum_{x_t \in \text{vocal}} \log p(x_t \mid \lambda_{s,i}) \right\} \tag{4.2}$$

For implementation efficiency, GMMs with diagonal covariance matrices are used throughout this project.

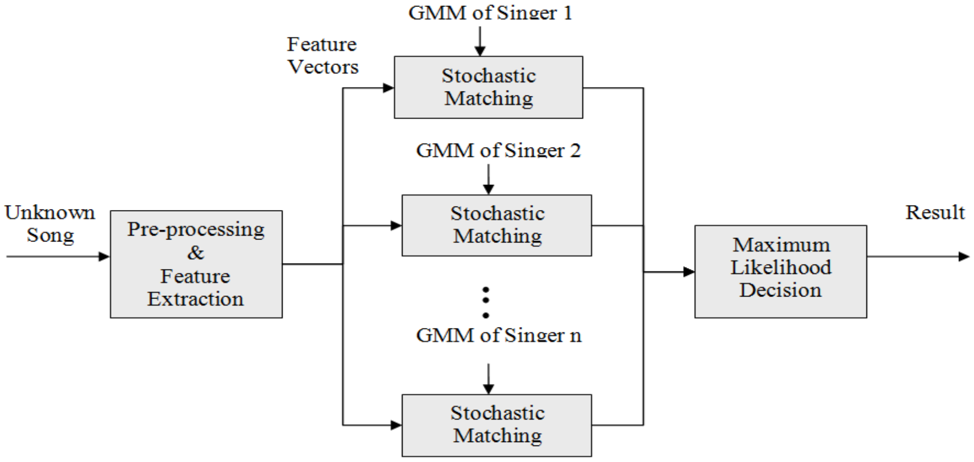

Figure 4.7 Identification with GMM

Frame based decision is applied for the other two classifiers. It identifies the singer which is singing that frame. This is done for all the frames of unknown song. Then, as a result, a singer is decided for the unknown song which singer is selected mostly.

For linear method, a decision tree has to be constructed because of SVM makes a binary classification and there are more than two singer groups. We design a decision tree for identification. The details of it are given in Figure 4.7.

In this decision tree, N different SVM classifier is trained where N is the number of singers. For training data of each classifier, a group of songs from which singer is going to be classified and a group of songs from other singers are used. After training, the frame from unknown song is classified whether it belongs to 1st singer or not. If not, it is classified whether it belongs to 2nd singer or not. If it is not classified as any of the singers, then this frame is identified as an undefined frame.

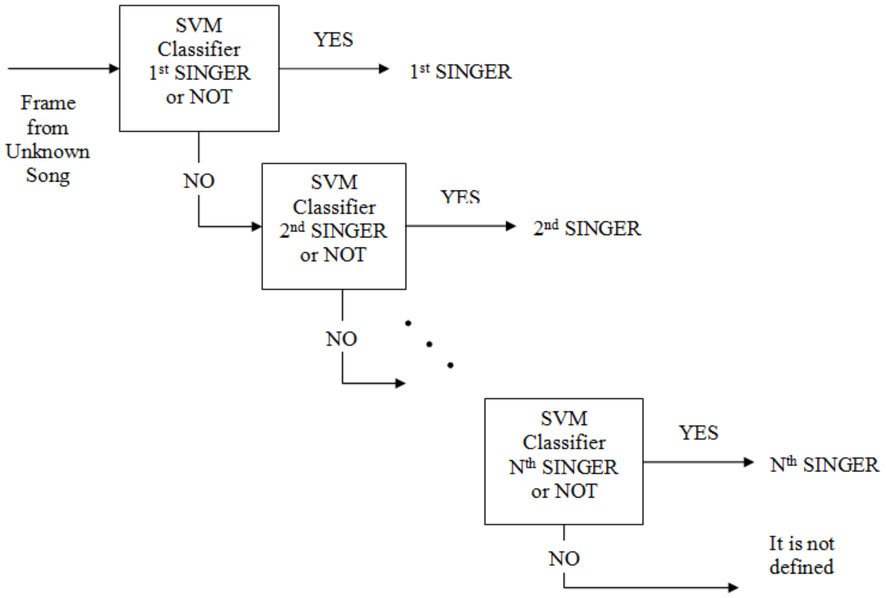
SVM Classifier 1st SINGER or NOT
YES
1st SINGER
Frame from Unknown Song
NO
SVM Classifier 2nd SINGER or NOT
YES
2nd SINGER
NO
SVM Classifier Nth SINGER or NOT
YES
Nth SINGER
NO
It is not defined

Figure 4.7 Identification with SVM and constructed decision tree

Lastly, we use a MLP network for identification. It is more complicated compared to statistical or linear classifier in terms of decision. We simply assign an output neuron to each singer with hardlimit transfer function. In theory, when a frame is given to this network, the output of the neuron, which is assigned to the singer of song that frame taken, is 1, rest is 0. If the output is 1 for more than one neuron, then this frame is not used. The system using MLP network with this model is shown in Figure 4.8.

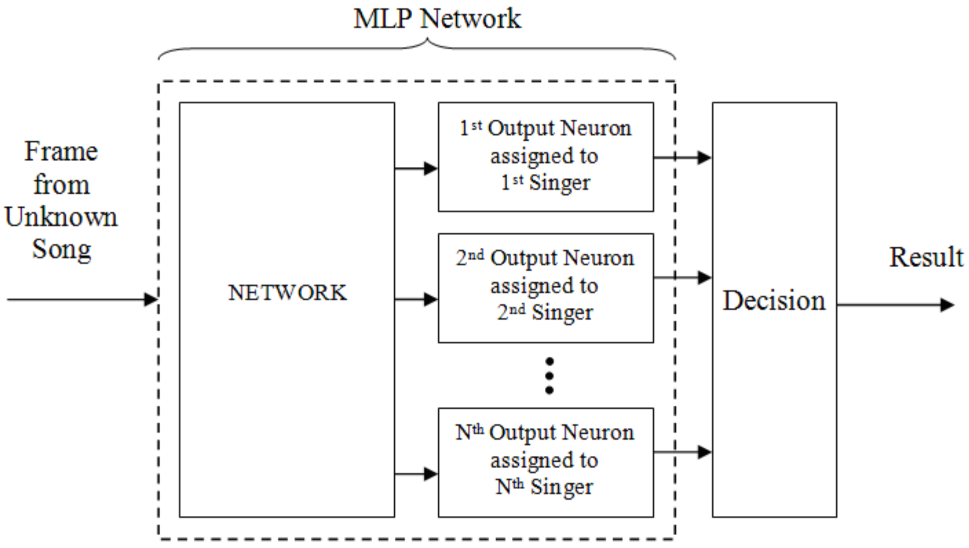
MLP Network
Frame from Unknown Song
NETWORK
1st Output Neuron assigned to 1st Singer
2nd Output Neuron assigned to 2nd Singer
Nth Output Neuron assigned to Nth Singer
Decision
Result

Figure 4.8 Identification with MLP network

CHAPTER FIVE
EXPERIMENTAL RESULTS

Up to now, all implementations covered by this project are presented. The experimental results will follow with this chapter. While doing experiments, the aim is getting the best performance for segmentation and identification phase of SID problem. Therefore, many experiments are performed depending on all the parameters of features and classifiers. There are lots of parameters which may be very important. Some of them are also neglected which we have not significant enough effect on the results.

One of the most important parameters is the frame size. This is kept ~1s for energy, zero crossing rate and spectral flux, and ~30ms for cepstral coefficients. All the results given below are done with these frame sizes. For the cepstral coefficients, we observe the results by changing the number of coefficients. The results are listed selecting only some of them which have more influence over the results. The features combinations and statistical parameters such as mean, variance and skewness are also used but they are not listed since they do not have observable effects on the accuracy.

For classifiers, we again perform the experiments by changing the parameters which have the greatest influence on the results. The results listed below include GMM results depending on the number of gaussians, SVM results depending on kernel functions, and MLP results depending on the methods given in previous chapter.

Additionally, we use hamming window for windowing and ~70% shift for overlapping. As mentioned in previous chapter, diagonal covariance matrices are used for implementation efficiency of GMM classifiers, and the network is constructed with one hidden layer which has 8-12 neurons for MLP network. Inputs are defined according to feature vector size and gradient descent algorithm is used for back propagation.

5.1 Segmentation Results

There are three classifiers and three different feature sets for segmentation part. Results are presented according to classifiers. First feature set is constructed with combinations of energy, spectral flux and zero crossing rate. Other two feature sets are constructed with different number of coefficients.

5.1.1 GMM Results

The most important parameter for GMM is number of Gaussian used. The experiments are done with 4, 8, 16, 32 and 64 Gaussians.

Table 5.1 The percentage results of correct vocal (V) and non-vocal (N) classification for GMM classifier with using three different feature sets

Features		Number of Gaussians in GMM									
		4		8		16		32		64	
		V	N	V	N	V	N	V	N	V	N
Energy, Spectral Flux, Zero Crossing Rate	E	62	56	64	65	65	64	58	64	59	63
	ZCR	53	24	57	26	58	31	61	27	60	29
	SF	59	45	61	43	62	47	65	52	64	53
	E + SF	67	59	67	57	70	58	65	63	58	62
	E + ZCR	65	52	64	49	67	51	61	51	59	53
	SF + ZCR	57	37	61	34	62	37	57	41	49	43
	E + SF + ZCR	71	58	74	54	75	56	70	51	65	48
MFCC	4 Coefficients	74	69	74	72	77	74	79	75	80	77
	8 Coefficients	77	75	79	74	80	76	83	79	84	82
	12 Coefficients	81	77	83	80	84	82	87	83	89	84
	16 Coefficients	84	79	88	87	89	88	88	85	88	86
	20 Coefficients	86	83	94	93	92	89	89	87	90	86
	24 Coefficients	88	81	92	91	92	89	90	89	87	88
	28 Coefficients	84	80	89	90	90	89	91	90	90	90
LPCC	4 Coefficients	71	57	72	60	76	59	74	65	75	59
	8 Coefficients	74	61	74	63	79	65	77	66	76	64
	12 Coefficients	82	66	83	65	82	68	81	68	80	67

The best results, 94% for vocal part and 93% for non-vocal part, are got with 8 gaussians using 20 MFCC shown in Table 5.1 for segmentation using GMM classifier.

5.1.2 SVM Results

The most important parameter for SVM is kernel function used for defining hyper plane. The experiments are done with three kernel functions such linear, quadratic ad radial basis.

Table 5.2 The percentage results of correct vocal (V) and non-vocal (N) classification for SVM classifier with using energy (E), zero crossing rate (ZCR) and spectral flux (SF) as features

Features		Linear Kernel		Quadratic Kernel		RBF Kernel	
		V	N	V	N	V	N
Energy, Spectral Flux, Zero Crossing Rate	E	61	25	65	29	49	24
	ZCR	59	63	71	75	81	83
	SF	66	75	51	68	35	58
	E + SF	69	35	83	74	65	66
	E + ZCR	86	69	87	71	75	68
	SF + ZCR	55	60	78	74	74	72
	E + SF + ZCR	74	61	79	75	83	72
MFCC	4 Coefficients	57	60	61	56	69	68
	8 Coefficients	56	52	59	41	68	64
	12 Coefficients	54	45	62	39	79	64
	16 Coefficients	59	46	63	48	84	67
	20 Coefficients	59	46	61	42	89	75
	24 Coefficients	61	47	61	43	90	77
	28 Coefficients	62	51	59	41	91	83
LPCC	4 Coefficients	62	63	59	60	67	63
	8 Coefficients	65	57	65	63	65	63
	12 Coefficients	68	52	67	65	64	64

The best results, 91% for vocal part and 83% for non-vocal part, are got with radial basis kernel using 28 MFCC shown in Table 5.2 for segmentation using SVM classifier.

5.1.3 MLP Results

There are a lot of parameters for MLP classifier such back propagation algorithm, number of hidden layers and neurons in each hidden layer, output layer for decision. The gradient descent algorithm is used. The networks are constructed with one hidden layer and 8 neurons in hidden layer. Experiments are done with two methods mentioned in Chapter 4.

Table 5.3 The percentage results of correct vocal (V) and non-vocal (N) classification for MLP classifier with using energy(E), zero crossing rate (ZCR) and spectral flux(SF) as features

Features		First Method		Second Method	
		V	N	V	N
Energy, Spectral Flux, Zero Crossing Rate	E	53	47	45	46
	ZCR	36	29	37	30
	SF	48	46	47	44
	E + SF	55	58	49	51
	E + ZCR	52	49	47	41
	SF + ZCR	50	41	44	32
	E + SF + ZCR	61	54	57	49
MFCC	4 Coefficients	72	73	64	64
	8 Coefficients	75	73	67	67
	12 Coefficients	79	75	70	68
	16 Coefficients	81	76	72	71
	20 Coefficients	83	79	76	73
	24 Coefficients	84	81	77	75
	28 Coefficients	87	84	78	76
LPCC	4 Coefficients	71	62	63	55
	8 Coefficients	74	66	68	59
	12 Coefficients	76	69	72	64

The best results, 87% for vocal part and 84% for non-vocal part, are got with first method using 28 MFCC shown in Table 5.3 for segmentation using MLP classifier.

5.2 Identification Results

There are three classifiers and two different feature sets for identification part. Results are presented according to classifiers. Feature sets are constructed with different number of coefficients.

5.2.1 GMM Results

The experiments are done with 8, 16, 24, 32 and 64 Gaussians. The best result, 91%, is got with first method using 20 MFCC shown in Table 5.4 for identification using GMM classifier.

Table 5.4 The percentage results of correct singer classification for GMM classifier with using MFCC as features

Features		8 Mixture	16 Mixture	24 Mixture	32 Mixture	64 Mixture
MFCC	8 Coefficients	74	81	80	80	78
	12 Coefficients	75	82	81	82	83
	16 Coefficients	84	88	86	86	80
	20 Coefficients	86	91	90	89	87
	24 Coefficients	86	88	88	86	86
	28 Coefficients	84	88	88	86	85
	8 Coefficients	66	73	71	64	68
	12 Coefficients	71	74	76	75	71

LPCC

5.2.2 SVM Results

The experiments are done again with three different kernel functions. The best result, 86%, is got with radial basis kernel using 24 MFCC shown in Table 5.5 for identification using SVM classifier.

Table 5.5 The percentage results of correct singer classification for SVM classifier with using MFCC as features

Features		Linear Kernel	Quadratic Kernel	RBF Kernel
MFCC	8 Coefficients	63	69	74
	12 Coefficients	67	67	77
	16 Coefficients	70	74	82
	20 Coefficients	76	81	85
	24 Coefficients	74	76	86
	28 Coefficients	71	79	82
LPCC	8 Coefficients	62	64	72
	12 Coefficients	67	69	75

5.2.3 MLP Results

The networks are constructed with one hidden layer and 12 neurons in hidden layer. The best result, 71%, is got using 12 LPCC shown in Table 5.6 for identification using MLP classifier.

Table 5.6 The percentage results of correct singer classification for MLP classifier with using MFCC as features

Features		Result
MFCC	8 Coefficients	59
	12 Coefficients	63
	16 Coefficients	61
	20 Coefficients	66
	24 Coefficients	68
	28 Coefficients	65
LPCC	8 Coefficients	67
	12 Coefficients	71

5.3 SID Results

The features and classifiers which have the best performance over all the experiments are selected for the complete SID system. 20 MFCCs are used for both segmentation and identification. For classifier, 8-mixture GMM is used for segmentation and 16-mixture GMM is used for identification. Proposed SID system is shown in detail with Figure 5.1.

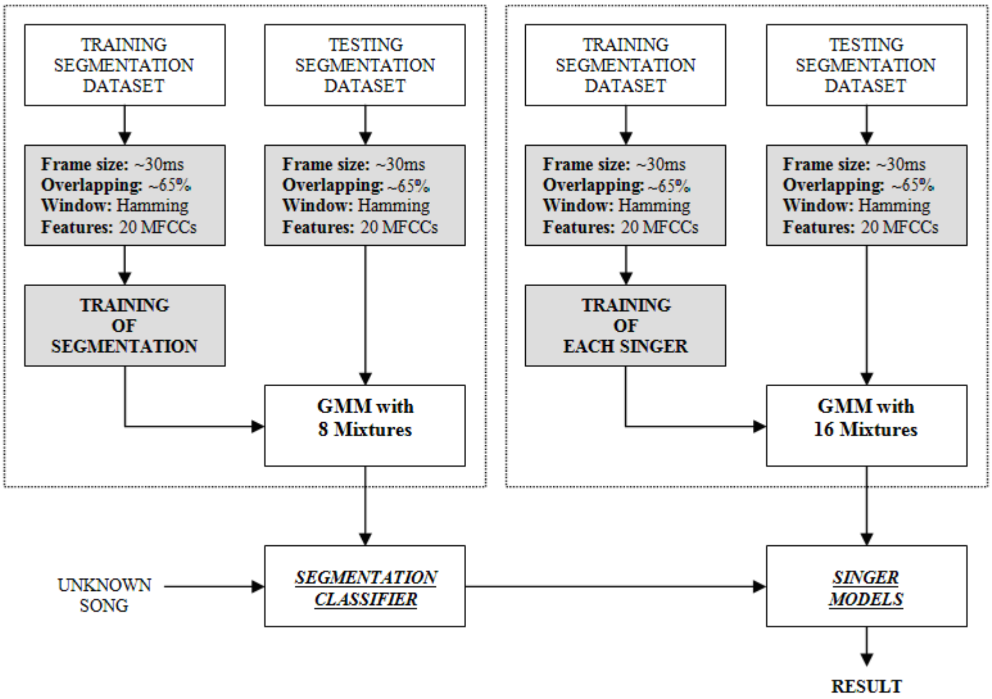

Figure 5.1 Block diagram of the SID system

The overall accuracy is 83.3%. According to this system, the confusion matrix is given in Table 5.7.

Table 5.7 The confusion matrix of the system

	S01	**S02**	**S03**	**S04**	**S05**	**S06**	**S07**	**S08**	**S09**	**S10**	**S11**	**S12**	**S13**	**S14**	**S15**
S01	8														
S02		7													
S03			10												
S04				10											
S05		1			10	1			1	1	1				
S06						8									
S07						1	9								
S08		1						9	1		2	1			
S09									7						
S10										5					
S11											5				1
S12												9			
S13	2	1					1		1	2			9		
S14								1			1		1	10	
S15										2	1				9

According to confusion matrix, the sensitivity and specifity values of all the singers are shown in Table 5.8.

Table 5.8 The performance table of the system

Singer	Sensitivity	Specifity
S01	80	100
S02	70	100
S03	100	100
S04	100	100
S05	100	97.3
S06	80	100
S07	90	99.3
S08	90	95.3
S09	70	100
S10	50	100
S11	50	99.3
S12	90	100
S13	90	95.3
S14	100	100
S15	90	97.3

CHAPTER SIX
CONCLUSION

In this thesis, we have presented a set of effective methods for the automatic identification of singers from various musical genres and constructed a SID system with the methods gives the best results over all the experiments. We hope that our efforts will prove to be useful in the future as additional research in this and related areas are undertaken.

The basis of the system is identifying the singer of a song with singer models trained by songs of singers. Therefore, the data will be used for singer models should belong to singer as much as possible. Eliminating the non-vocal parts from the songs is very important. For that reason, the system consists of two steps and each one is a challenging problem for MIR applications.

As it is presented in previous chapters, the solution for each problem consists of feature extraction and supervised classification using these features. There are a lot of parameters should be defined for feature extraction and classification parts and all of them are almostly an optimization problem. We have discarded some of the parameters and done the experiments in ranges for rest of them considering the results of done works.

For regimentation problem, it is aimed to get the singer voice as much as possible. However the spectral and temporal similarities between the instruments' sounds and singer voice make it harder. In theory, the start of singing voice is normally reflected as a sudden rise in the energy level of the audio signal, and often indicated by the appearance of high peaks in the spectral flux value, because the voice signal tends to have higher rate of change than instrumental music. Also while zero crossing rate values of instrumental music are normally within a relatively small range, the singing voice is often indicated by high amplitude ZCR peaks resulted from pronunciations of consonants. Therefore, we used them as features for segmentation. The best accuracy achieved 87%. But the results are not as well as expected. This might be caused by using songs from different genres. Better results can be achieved if the experiments are done with only one genre. Our aim is to generalize the solution of the problem. Therefore, we have used cepstral coefficients which have good results for speech recognition and speaker identification applications as well. The MFCCs gives a 94% accuracy but LPCCs did not improve the results. For the classifiers, the best accuracies were achieved with SVM using energy and spectral flux as features and GMM using MFCC as features. The MLP gave worse results because of optimization of parameters and uncertainty of changing initial values.

For singer modeling, we made experiments with cepstral coefficients as features and GVM, SVM and MLP as classifiers. The aim of using cepstral coefficients is that the low order cepstral coefficients represent information about the vocal tract shape, and the high order coefficients characterize the source signal. Both parts contain important information about voice identity. We got the best results using MFCCs with GMM. It was about 91%. It has been proved that the statistical modeling is a effective method for MIR application. SVM had the worse results as it was expected. It achieved 86% accuracy. The reason is that SVM is a binary classifier and the decision is important. The purpose of decision in this work is easiest solution. More complex decision trees may give better results. MLP has the worst results as 71%. We did not get good performance from MLP because of the same reason mentioned for segmentation. Also it requires more computational time for training.

The final SID system has been constructed with GMM classifiers. It achieved an 83.3% accuracy. The results we got when testing the final SID system were not good as expected, but the most important purpose of this project is to analyze the SID problem.

For future directions, feature selection algorithms can be applied to choose the best features. New feature sets and new classifiers can be combined for better performance. For segmentation part, some lyrics detection systems therefore used for karaoke application might be applied to get more effective features for singer modeling.

REFERENCES

Allegro, P., (2008). *Singing Voice Detection in Polyphonic Music Signals*. Retrieved January 04, 2009, from http://paginas.fe.up.pt/~ee02193/files.php

B. Whitman, G. Flake, and S. Lawrence. (2001). Artist detection in music with minnowmatch. *Proc. 2001 IEEE Workshop on Neural Networks for Signal Processing*, 559–568.

Bartsch, M. A., & Wakefield, G. H. (2004). Singing voice identification using spectral envelope estimation. *IEEE Transactions on Audio, Speech, and Language Processing*, 12 (2), 100-109.

Berenzweig, A. L., and Ellis, D. P. W. (2001). Locating Singing Voice Segments within Music Signals. *Proc. IEEE Workshop Applications of Signal Processing to Audio and Acoustics*, 119–122.

Berenzweig, A., Ellis, D. P. W., & Lawrence, S. (2002). Using voice segments to improve artist classification of music. *AES 22th International Conference on Virtual, Synthetic and Entertainment Audio*.

Duda, R. O., Hart, P. E., Stork, D. G. (2001). *Pattern classification* (2nd ed.). Canada: Wiley.

Ellis, D. P. W. (2007). *Classifying Music Audio with Timbral and Chroma Features*. Retrieved April 22, 2008, from http://www.ee.columbia.edu/~dpwe/pubs/

Feng, L., Nielson, A. B., and Hansen, L. K., (2008). Vocal Segment Classification in Popular Music. *9th International Conference on Music Information Retrieval.*

Fujihara, H., Kitahara, T., Goto, M., Komatani, K., Ogata, T., & Okuno, H. G. (2005). Singer identification based on accompaniment sound reduction and reliable frame selection. *6th International Conference on Music Information Retrieval.*

Kim, Y. E. (2003). Singer identification and transformation through dynamic modelling of vocal fold and vocal tract parameters. *Proceedings of the 2003 Stockholm Music Acoustics Conference*.

Maddage, N. C., Xu, C., & Wang, Y. (2004). Singer identification based on vocal and instrumental models. *Proceedings of the 17th International Conference on Pattern Recognition*.

Mesaros, A., & Astola, J. (2005). The mel-frequency cepstral coefficients in the context of singer identification. *6th International Conference on Music Information Retrieval.*

Mesaros, A., Virtanen, T., & Klapuri, A. (2007). *Singer identification in polyphonic music using vocal separation and pattern recognition methods*. Retrieved May 11, 2008, from http://www.cs.tut.fi/sgn/arg/

Nwe, T. L., & Li, H. (2007). Exploring vibrato-motivated acoustic features for singer identification. *IEEE Transactions on Audio, Speech, and Language Processing*, 15 (2), 330-340.

Reynolds, D. A., and Rose, R. C., (1995). Robust Text-independent Speaker Identification Using Gaussian Mixture Speaker Models. *IEEE Trans. Speech Audio Process.*, Vol.3, 72-83

Tsai, W. H., & Wang H. M. (2006). Automatic singer recognition of popular music recordings via estimation and modeling of solo vocal signals. *IEEE Transactions on Audio, Speech, and Language Processing*, 14 (1), 330-340.

Tsai, W. H., Wang, H. M. & Rodgers, D. (2003). Automatic Singer Identification of Popular Music Recordings via Estimation and Modeling of Solo Vocal Signal. *Proc. 8th Eur. Conf. Speech Communication and Technology*.

Zhang, T. (2003). *System and Method for Automatic Singer Identification*. Retrieved June 21, 2007, from http://www.hpl.hp.com/techreports/

APPENDICES

The list of songs and singers:

	S01 – Aerosmith	**S02 – Beatles**	**S03 - Dave Matthews**
01	young lust	no reply	so much to say
02	fine	i’m a loser	two step
03	love in an elevator	baby’s in black	crash into me
04	monkey on my back	rock and roll music	too much
05	janie’s got a gun	i’ll follow the sun	say goodbye
06	the other side	mr. moonlight	drive in drive out
07	my girl	eight days a week	let you down
08	don’t get mad get even	words of love	lie in our graves
09	voodoo medicine man	every little thing	cry freedom
10	what it takes	what you’re doing	tripping billies
11	same old song and dance	magical mystery tour	i did it
12	lord of the thighs	the fool on the hill	the space between
13	spaced	flying	dreams of our fathers
14	woman of the world	blue jay way	so right
15	s.o.s.	i am the walrus	if i had it all
16	train kept a rollin	hello goodbye	what you are
17	pandora’s box	strawberry fields forever	angel
18	draw the line	penny lane	fool to think
19	i wanna know why	baby you’re a rich man	sleep to dream her
20	critical mass	all you need is love	dreamgirl
21	get it up	taxman	stand up for it
22	bright light fright	eleanor rigby	american baby
23	kings and queens	yellow submarine	smooth rider
24	the hand that feeds	she said she said	out of my hands
25	sight for sore eyes	good day sunshine	hello again
26	milk cow blues	for no one	louisiana bayou
27	back in the saddle again	doctor robert	stolen away on 55th 3rd
28	last child	i want to tell you	you might die trying
29	rats in the callar	and your bird can sing	steady as we go
30	combination	tomorrow never knows	hunger for the great light

	S04 - Depeche Mode	**S05 - Garth Brooks**	**S06 - Green Day**
01	new life	the thunder rolls	armatage shanks
02	puppets	new way to fly	brat
03	boys say go	victim of the game	stuck with me
04	nodisco	friends in low places	geek stink breath
05	what’s your name	this ain’t tennessee	no pride
06	photographic	wild horses	86
07	big muff	unanswered prayers	panic song
08	just can’t get enough	same old story	brain stew
09	dreaming of me	mr. blue	westbound sign
10	ice machine	wolves	tight wad hill
11	something to do	against the grain	nice guys finish last
12	lie to me	rodeo	hitchin a ride
13	people and people	what she’s doing now	the grouch
14	it doesn’t matter	burning bridges	redundant
15	stories of old	which one of them	scattered
16	somebody	papa loved mama	all the time
17	master and servant	shameless	worry rock
18	if you want	cold shoulder	uptight
19	blasphemous rumours	in lonesome dove	last ride in
20	the things you said	the river	walking alone
21	strangelove	we shall be free	warning
22	sacred	mr. right	church on sunday
23	little 15	every now and then	fashion victim
24	behind the wheel	walking after midnight	castaway
25	i want you now	dixie chicken	misery
26	to have and to hold	learning to live again	deadbeat holiday
27	nothing	that summer	hold on
28	pimpf	something with a ring to it	jackass
29	agent orange	night rider’s lament	waiting
30	pleasure little treasure	face to face	minority

	S07 – Led zeppelin	**S08 – Madonna**	**S09 – Metallica**
01	good times bad times	he's a man	the house jack built
02	you shook me	sooner or later	until it sleeps
03	dazed and confused	hanky panky	king nothing
04	your time is gonna come	i'm going bananas	hero of the day
05	babe i'm gonna leave you	cry baby	bleeding me
06	black mountain side	something to remember	cure
07	i can't quit you baby	back in business	poor twisted me
08	how many more times	more	wasting my hate
09	whole lotta love	what can you lose	mama said
10	the lemon song	vogue	thorn within
11	thank you	like a prayer	the outlaw torn
12	heartbreaker	express yourself	enter sandman
13	ramble on	love song	sad but true
14	moby dick	promise to try	holier than thou
15	bring it on home	cherish	the unforgiven
16	friends	dear jessie	wherever i may roam
17	celebration day	oh father	don't tread on me
18	out on the tiles	keep it together	through the never
19	gallows pole	spanish eyes	nothing else matters
20	tangerine	act of contrition	of wolf and man
21	that's the way	drowned world	the god that failed
22	immigrant song	ray of light	my friend of misery
23	black dog	candy perfume girl	the struggle within
24	rock and roll	skin	the memory remains
25	the battle of evermore	nothing really matters	devil's dance
26	stairway to heaven	sky fits heaven	better than you
27	misty mountain hop	frozen	carpe diem baby
28	four sticks	the power of good bye	low man's lyric
29	going to california	little star	attitude
30	when the levee breaks	mer girl	fixxxer

	S10 - Queen	**S11 - Radiohead**	**S12 - Roxette**
01	brighton rock	planet telex	church of your heart
02	killer queen	the bends	small talk
03	tenement funster	high and dry	phsycial fascination
04	flick of the wrist	fake plastic trees	perfect day
05	lily of the valley	bones	the look
06	now ı m here	nice dream	dressed for success
07	ın the lap of the gods	just	sleeping single
08	stone cold crazy	my ıron lung	paint
09	dear friends	black star	dance away
10	misfire	sulk	cry
11	we will rock you	you	chances
12	we are the champions	creep	dangerous
13	sheer heart attack	how do you	view from a hill
14	all dead all dead	stop whispering	shadow of a doubt
15	spread your wings	thinking about you	listen to your heart
16	fight from the ınside	ripcord	soul deep
17	who needs you	vegetable	secrets that she keeps
18	ıt s late	prove yourself	goodbye to you
19	my melancholy blues	ı can t	ı call your name
20	mustapha	lurgee	surrender
21	fat bottomed girls	blow out	voices
22	jealousy	creep radio edit	neverending love
23	bicycle race	airbag	call of the wild
24	ıf you can t beat them	exit music for a film let down	joy of a toy
25	let me entertain you	karma police	like lovers do
26	dead on time	fitter happier	so far away
27	ın only seven days	electioneering	pearls of passion
28	dreamers ball	climbing up the walls	turn to me
29	fun ıt	no surprises	church of your heart
30	leaving home ain t easy	lucky	small talk

	S13 - Steely Dan	**S14 - Tori Amos**	**S15 - U2**
01	babylon sisters	pretty good year	zooropa
02	hey nineteen	bells for her	babyface
03	glamour profession	past the mission	numb
04	time out of mind	baker baker	lemon
05	my rival	the wrong band	stay faraway so close
06	third world man	the waitress	some days are better than others
07	bodhisattva	cornflake girl	the first time
08	razor boy	ıcicle	dirty day
09	the boston rag	cloud on my tongue	the wanderer
10	your gold teeth	space dog	sunday bloody sunday
11	show biz kids	yes anastasia	seconds
12	my old school	spring haze	new year s day
13	pearl of the quarter	1000 oceans	like a song
14	king of the world	parasol	drowning man
15	black cow	sweet the sting	the refugee
16	parker s band	jamaica ınn	two hearts beat as one
17	through with buzz	barons of suburbia	red light
18	pretzel logic	sleeps with butterflies	surrender
19	kid charlemagne	general joy	40
20	the caves of altamira	mother revolution	gloria
21	don t take me alive	ribbons undone	ı fall down
22	sign in stranger	cars and guitars	ı threw a brick through a window
23	the fez	witness	rejoice
24	green earrings	original sinsuality	fire
25	haitian divorce	ıreland	tomorrow
26	everything you did	the beekeeper	october
27	the royal scam	hoochie woman	with a shout
28	with a gun	goodbye pisces	stranger in a strange land
29	charlie freak	marys of the sea	scarlet
30	monkey in your soul	toast	ıs that all