BOOSTING CLASSIFIERS FOR AUTOMATIC MUSIC GENRE CLASSIFICATION

by

Ulaş Bağcı

A Thesis Submitted to the
Graduate School of Engineering
in Partial Fulfillment of the Requirements for
the Degree of

Master of Science

in

Electrical & Computer Engineering

Koç University

September,2005

Koç University

Graduate School of Sciences and Engineering

This is to certify that I have examined this copy of a master's thesis by

Ulaş Bağcı

and have found that it is complete and satisfactory in all respects,
and that any and all revisions required by the final
examining committee have been made.

Committee Members:

_____

Assist. Prof. Engin Erzin

_____

Assist. Prof. Yücel Yemez

_____

Assist. Prof. Hakan Erdoğan

Date: _____

To My Family

ABSTRACT

Music genre classification is an important tool for music information retrieval systems and has been finding important applications in various media platforms. Two important problems of the automatic music genre classification are feature extraction and classifier design. There are recent works on these problems with promising future research directions. This thesis investigates discriminative boosting of classifiers to improve the automatic music genre classification performance. Two-class of classifiers, boosting of the Gaussian mixture model based classifiers and classifiers that are using the inter-genre similarity information, are proposed. Boosting is a technique that combines sequentially trained classifiers, where in each new classifier a better modeling of hard-to-classify samples is done, and the overall performance is boosted in the combined classifier. In this thesis a novel extension is proposed to the maximum-likelihood based training of the Gaussian mixtures to integrate GMM classifier into boosting architecture. Later, the boosting idea is modified to better model the inter-genre similarity information over the mis-classified feature population. Once the inter-genre similarities are modeled, elimination of the inter-genre similarities reduces the inter-genre confusion and improves the identification rates. Finally, an auto-clustering scheme is build to determine similar music genre types for hierarchical classifier structure. Experimental results with promising identification improvements are provided.

ÖZETÇE

Müzik türlerinin sınıflandırılması, müzik bilgi erişimi sistemlerinde ve farklı medya ortamlarında önemli bir araç olarak kullanılmaktadır. Öznitelik çıkarma ve sınıflandırıcı tasarımı müzik türlerinin sınıflandırılmasında iki onemli problem olarak karşımıza çıkmaktadır. Bu tezde, istatistiksel olarak daha iyi sınıflandırıcı oluşturabilmek icin Gauss karışımı modelli (GMM) sınıflandırıcıların farklılıklar gözetilerek yükseltilmesi (Boosting) ve müzik türleriarası benzerliklerini(IGS) kullanarak yükseltme algoritmasına alternatif bir yöntem sunulmaktadir. Sınıflandırıcı yükseltilmesi, her bir eğitilmiş sınıflandırıcıda sınıflandırılmasi zor olan örneklerin yeniden modellenmesi ve yeni modellenen bu sınıflandırıcıların ardısık olarak birleştirilmesi ile sağlanır. Bu tezde, gauss karışım modellerinin en büyük olabilirlik kestirimine dayanan eğitim yönteminin sınıflandırıcı yükseltme yapısına uyumunu sağlayan yeni bir teknik sunulmaktadir. Sınıflandırıcı yükseltilmesi düsüncesinden yola cıkarak geliştirdiğimiz IGS düzeneğinde ise, benzer kesişim öznitelik vektörleri olarak adlandırdığımız yanlış sınıflandırılan vektörler biraraya geldikten sonra gauss karışımlarıyla yeniden modellenir ve müzik türleri arasındaki bozulma oranının düsük olması için test aşamasında yanlış tanınan örnekler sistemden atılır. Tezin son kısmında ise müzik türlerinin birbirine benzerliklerini Euclidian-uzaklığı esas alarak otomatik bir şekilde bulan bir yöntem geliştirilmiştir.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

NOMENCLATURE

DA	Database A
DB	Database B
DSP	Digital Signal Processing
DWCH	Daubechies Wavelet Coefficient Histogram
EM	Expectation Maximization
FFT	Fast Fourier Transform
GMM	Gaussian Mixture Model
HAC	Hierarchical Auto-Clustering
HACIGS	Hierarchical Auto-Clustering and Inter-Genre Similarity
IGS	Inter-Genre Similarity
IGD	Inter-Genre Differences
KNN	K-Nearest Neighbor
LDA	Linear Discriminant Analysis
LPC	Linear Predictive Coefficients
LS	Least-Squares
MFCC	Mel-Frequency Cepstral Coefficient
ML	Machine Learning
MSP	Music Signal Processing
SVM	Support Vector Machines
STFT	Short-Time Fourier Transform
TTA	First Scenario in Testing and Training
TTB	Second Scenario in Testing and Training
VQ	Vector Quantization
ZCR	Average Zero-Crossing Rate

cl	Classical music
co	Country music
di	Disco music
hi	Hip-hop music
ja	Jazz music
me	Metal music
po	Pop music
re	Reggae music
ro	Rock music
pdf	Probability Density Function

ACKNOWLEDGMENTS

I would like to thank my supervisor Assist. Prof. Engin Erzin who has been a great source of inspiration and provided the right balance of suggestions, criticism, and freedom.

I also would like to thank George Tzanetakis for sharing his musical genre database.

Chapter 1

INTRODUCTION

"Music is a higher revelation than all wisdom and philosophy. Music is the electrical soil in which the spirit lives, thinks and invents."

Ludwig van Beethoven

The Music is defined as an artistic form of auditory communication incorporating instrumental or vocal tones in a structured and continuous manner, more generally the music bases on the special interactions between sounds. In the history, people have made their music and confronted with music everywhere. Perception of music varies from person to person, and it has been known that, even for humans, music genre perception is a hard task.

The concept of musical genre is a dual one. On one hand, genre is an interpretation of a music piece through a cultural background. It is a linguistic category providing common term to talk about the music. On the other, it is a set of music pieces sharing similar properties of music, like tempo and instrumentation. Musical genres themselves are ill-defined; attempts to define them precisely will most probably end up in circular structures [1].

A music genre is a category of pieces of music that share certain styles or characteristics. Actually, the definition of music genre is not restricted to only content of the music. The non-musical content such as the geographical origin of the music can be a measure to characterize the music as a specific genre. Genre may be used as an intentional concept. In this view, genre is an interpretation of a title, produced and possibly shared by a given community, much in the same way we arrogate and interpret meanings to words in our languages. Alternatively, genre may be used as an extensional concept. In this view, genres are sets of music titles. Here, genre is closely related to analysis, where it is a dimension of music title, much like tempo, timbre or the language of the lyrics. In idealistic, mathematical

words, intentional and extensional definitions coincide. In the real world they do not, so no unique position can be taken regarding genre [1]. In this thesis we will analyze only the musical genre as an expressive style and characteristics of music defined with differences of instrumentation, rhythmic and harmonic content of the music.

1.1 Music Genre Perception of Human

Listening experiments in which human abilities to recognize musical genres and proposed methods are compared in many studies. The experiments show that people without any formal musical training can quite accurately and easily classify music into musical genres, even from short excerpts. In general, people can easily make a clear distinction for instance between classical music and rock music. However, musical genres do not have clear boundaries or definitions since music is an art form that evolves constantly. A musical genre can be considered always to be a subjective category with regard to both listeners and the cultural background. For realistic study, it is necessary to compare human decision results with machine-made results.

There have not been many studies concerning the human ability to classify music genres. One of the studies was conducted by Perrot in [2]. For three-second samples subjects scored 70% accuracy compared to the record companies' original music genre classes. For 250 ms samples the recognition accuracy was still around 40%. These results are quite interesting because they show that humans can, in fact, recognize music genres without using any higher-level abstractions, like rhythm. The shortest 250 ms excerpts are by far too short to enable perception of the rhythm, melody or to comprehend the structure of music. Hence, short-time features such as spectral and temporal features are necessary for automatic genre recognition. Rhythmic and melody features can be utilized for longer excerpts to further improve the music genre recognition.

1.2 Genre Classification

Music genre classification is crucial for the categorization of bulky amount of music content. Automatic music genre classification finds important applications in professional media production, radio stations, audio visual archive management, entertainment and recently on the Internet. Especially in professional media production, need to interact with large

audio collection increases in the music industry. Record company catalogues, record shops and megastores, music charts, musical web sites and online record shops, radios and TV stations are the main areas where genre classification is need to be used.

Although music genre classification is done mainly by hand and it is hard to precisely define the specific content of a music genre, it is generally agreed that audio signals of music belonging to the same genre contain certain common characteristics since they are composed of similar types of instruments and having similar rhythmic patterns. These common characteristics motivated recent research activities to improve automatic music genre classification [3, 4, 5, 6]. The problem is inherently challenging as the human identification rates after listening to 3s samples are reported to be around 70% [2].

Two important problems of automatic music genre classification are feature extraction and classifier design. Although we present an overview of the state-of-art music genre feature extraction methods, the main contribution of this thesis is building classifiers to boost music genre classification.

1.3 System Overview and Contribution

Numerous methods have been proposed for Music Genre Classification, but there is still a significant room to improve performance of automatic genre recognition. In this thesis, we designed discriminative classifiers to improve the automatic music genre classification rates. Two alternative classifier structures are proposed: i) Boosting of the Gaussian mixture model based classifiers, and ii) Classifiers that are using the inter-genre similarity (IGS) information. Using IGS with different modeling approaches such as flat distribution and hierarchical distribution show us the new IGS method is alternative of the Boosting classifiers and better than those combined classifiers from many computational perspectives. This work aims at providing quantitative answers to the following open questions:

- Can Boosting, collection of weak classifiers, be a way in improving performance of music genre classification?

- Can we employ inter-genre similarity to classify music genres better?

- Can hierarchical genre taxonomy be utilized to IGS for better classification of music

genres?

1.4 Organization

This thesis is divided into 6 chapters. Chapter 2 presents the literature review on music genre classification. In chapter 3, music signal feature extraction including timbral and rhythmic features are discussed. Chapter 4 presents music genre classification. In this chapter, after presenting a brief introduction to classifier design, the proposed novel music genre classification techniques are described.

Chapter 5 presents the experimental results of the proposed music genre classifiers with encouraging performance improvements over the state-of-art techniques. Chapter 6 summarizes the contributions to the concept of musical genre classification made in the thesis and suggests directions for further research.

Chapter 2

LITERATURE REVIEW

Feature extraction and classifier design are two important problems of the automatic music genre classification. An overview of the recent literature on these problems is given in the following sections.

2.1 Feature Extraction Literature

In the literature, features for music genre classification are examined in two groups: timbral and rhythmic content features. Timbral features characterize short-time properties, such as spectral information and zero-crossings. Rhythmic content features characterize long-term properties including beat, tempo, pitch content, etc. The instrumentation of a music performance has a significant influence in genre recognition. The timbre of constituent sound sources is reflected in the spectral distribution of a music signal. Thus spectral features are widely used in the literature. An extensive investigation of both timbral and rhythmic content features can be found in [6].

Mel-frequency cepstral coefficients (MFCC) have been dominantly used to model spectral information for speech recognition. In music genre classification, MFCC features are also widely used to represent timbral characteristics [6, 7, 8, 9]. Spectral information as a form of Fourier transform coefficients, MFCC features and linear prediction (LP) coefficients were used in [6]. Although LP has been designed to model speech production, it is also partially valid for for musical instruments. Since LP coefficients control the poles of transfer function, they characterize positions of the peaks on the power spectral density. Therefore, like MFCC features, LP coefficients can encode spectral envelope of the music signal. MFCC features combined with LP representation were used in [7, 9].

Wavelet Transform (WT) is a good time-frequency analysis tool. It provides high time resolution and low frequency resolution for high frequencies, and low time and high frequency resolution for low frequencies. In that respect it is similar to the human ear which exhibits

similar time-frequency resolution characteristics. The wavelet coefficients provide a compact representation that shows the energy of the signal in time and frequency. The distribution of energy in time and frequency for music is reported to be different from that of speech signal in [6, 10]. In another wavelet based approach, MPEG filterbank components are used to extract features, including centroid, flux, Roll-off and energy, directly from mpeg data [5, 6]. Feature extraction from the mpeg bit-stream losses some information during the process, so as to keep the information as much as possible a wavelet analysis based technique is used in feature extraction process.

Another novel feature extraction method is proposed in [3], in which audio signal is considered in two dimensional entity of the amplitude over time. Variations in the amplitude would be essential for music categorization, which is called distribution estimation in histogram technique. Decompositions of audio signals using wavelets produces a set of subband signals at different frequencies corresponding characteristics. As a wavelet filter, they used Daubechies wavelet filters Db8 with seven levels of decompositions. After decomposition, histogram of the wavelet coefficients are constructed at each subband. The histogram of the coefficients provides a good approximation of the wavelet variations at each subband. Using waveform distribution as a probability distribution function and taking first three moments of histogram combined with subband energy and average histogram provide discriminative feature sets called Daubechies wavelet coefficient histograms (DWCH).

One characteristic of music that has received considerable attention is the automatic extraction of the beat or tempo of musical signals. Informally the beat can be defined as an underlying regular sequence of pulses that coincides with the flow of music. Although this definition is vague, it gives a general idea of the beat characteristics. A more anthropocentric way of defining beat is as the sequence of human foot tapping when one listens to music. More elaborate discussions about the definition of beat can be found in Musicology and Music Cognition publications [11, 12]. Unlike typical automatic beat detection algorithms that provide only a running estimate of the main beat and its strength, beat histograms provide a richer representation of the rhythmic content of a musical piece, that can be used for similarity retrieval and classification. Extraction of beat histograms based on periodicity analysis of multiple frequency bands is described in [6, 10].

A common automatic beat detector structure consists of signal decomposition into fre-

quency bands using a filterbank, followed by an envelope extraction step and finally a periodicity detection algorithm which is used to detect the lags at which the signal's envelope is most similar to itself. The process of automatic beat detection resembles pitch detection with larger periods (approximately 0.5 seconds to 1.5 seconds for beat compared to 2 milliseconds to 50 milliseconds for pitch). The first three peaks of the enhanced autocorrelation function that are in the appropriate range for beat detection are selected and added to a beat histogram. The bins of the histogram correspond to beats-per-minute (bpm) from 40 to 200 bpm. For each peak of the enhanced autocorrelation function the peak amplitude is added to the histogram. That way peaks that have high amplitude (where the signal is highly similar) are weighted more strongly than weaker peaks in the histogram calculation. Unlike previous work in automatic beat detection which typically aims to provide only an estimate of the main beat (or tempo) of the song and possibly a measure of its strength, the beat histogram representation captures more detailed information about the rhythmic content of the piece that can be used to intelligently guess the musical genre of a song [10].

Some authors suggest that a genre classifier should not just rely on "global timbre" descriptors, but also take rhythm into account. Tzanetakis [6] uses a beat histogram built from the autocorrelation function of the signal: by looking at the weight of the different periodicities in the signal (and the ratios between these weights), one has an idea of the "strongness" and complexity of the beat in the music. This is an interesting feature to discriminate "straight-ahead" rock music from rhythmically complex world music, or classical music where the beat is not so accentuated.

Second order statistics such as angular second moment, correlation and entropy, which similarly account for time structure in the data are computed in [13]. Those are used also as features in genre recognition systems.

Soltau et al. uses a very specific set of features computed from the signal by a neural network in an unsupervised way [14, 15]. In a nutshell, it learns a set of abstract musical events from the tune. The input vector for the classifier is a vector of statistical measures on these abstract event: co-occurence of event i and j, etc. Thus, the system extracts information about the temporal structures of the songs. This coding may not be interpreted by human beings, but it is hopefully helpful for a machine learning algorithm to distinguish between genres [1].

Time envelope, low energy rate, RMS, loudness, central moments, predictivity ratio, beat strength and rhythmic regularity are the some other features used in the music genre classification. Among them beat strength and rhythmic regularity are the rhythm features, based on the statistical measure of histogram of beat.

Recently in [16], not only short time features are used but integration of short time features are presented as a novel technique. Medium time and long time features are defined to be used. Mean and the variance of the MFCCs are considered as medium type features. After calculating power spectrum of MFCC, it is summarized into four frequency bands, features in this application are called filterbank coefficients. High zero crossing rate ratio and low short-time energy ratio is also defined in the medium-time features. In long time features, many of the features at decision in long term have been derived from features at an earlier timescale. Beat spectrum and beat histogram are used both rhythmic feature type and long term feature type.

2.2 Music Genre Classification Literature

Various classifiers are employed for automatic music genre recognition including K-Nearest Neighbor (KNN) classifiers, Gaussian Mixture Models (GMM) classifiers and Support Vector Machine (SVM) classifiers.

Comparison of KNN and GMM based classifiersfor music genre recognition is studied in [3, 6]. In these works, a comprehensive set of features is extracted and explored using KNN and GMM classifiers. In [6] each class is trained with iterative EM to estimate the parameters of each Gaussian component and mixture weights. Then the KNN classifier is also used as a nonparametric classifier, where each sample is labeled according to the majority of its K nearest neighbors. KNN classifier has a simple architecture, which assigns the inputs to the class having the maximum number of samples among the K-neighbors belong to that class. Generally euclidean distance is used as a distance metric in KNN. Euclidean distances between test feature vector and all other train feature vectors are calculated and decision is made based on the minimum distance among the calculated ones. However, KNN rule is a suboptimal procedure, especially when $K = 1$, and its use will usually lead to an error rate greater than the minimum possible Bayes rate, but never worse than twice the Bayes rate [17]. For optimal KNN approach, we have to increase the value of K, number of neighbors.

As K increases, KNN will be optimal rather than suboptimal because lower bound on error rate will approach the Bayes rate. When K increases the computational complexity will increase both in space and time. Hence, KNN is not an efficient classifier for large database systems, such as music genre.

GMM with tree-based vector quantization for music genre classification is investigated in [5]. In tree-based vector quantization (TreeQ), after train data is partitioned into feature vectors, instead of modeling the acoustic directly, TreeQ trains a vector quantizer which segments the feature space into regions that have maximally different class populations. For each genre, histogram template can be extracted after the tree is constructed. Representation of the acoustic of a song or genre, histogram template, is the measure of quantity in each cell relatively. The cosine distance between any of two histogram templates provides an estimate of acoustic similarity [5]. It has some advantages like faster parameterization, but it is slow for user interactive applications.

Support Vector Machine (SVM) as in [3, 7, 18] is another well-known classifier technique, which is widely applied in many areas including speech and audio processing areas as well as music genre classification. In [3], the SVM is extended into multi-class classification with one-versus-the-rest, pairwise comparison, and multi-class objective functions. In the one-versus-the-rest method, binary classifiers are trained to separate one class from rest of the classes. The multi-class classification is then carried out according to the maximal output of these binary classifiers. In pairwise comparison, a classifier is trained for each possible pair of classes and the unknown observation is assigned to the class with the highest number of classification "votes" among all the classifiers. In the multi-class objective-function method, the objective function of a binary SVM is directly modified to allow the simultaneous computation of a multi-class classifier.

In [7], SVM is used as a multilayer classifiers to discriminate musical genres. In all of the layers SVM is used with different parameters and non-linear support vectors. In first layer music is classified into *pop/classic* and *rock/jazz* according to pre-defined features used in the SVM space. In the second layer, *pop/classic* music is further classified into *pop* and *classic* music and the process goes on like this. In each layer different kind of features are extracted and trained with SVM learning process, the parameters corresponding to three classifiers are different because the features used in each layers are different. Ogihara and

Li also used SVM in Music genre classification with taxonomy [18]. Unlike in the [7], they used linear kernels because in each layer they use the same feature space. SVM relies on preprocessing the data to represent patterns in a high dimension, typically higher than the original feature space, which can lead to limit computational resources [17]. In music genre classification, there are more than two classes, hence multi-categorical approach on SVM should be used. One of the technique to reduce multi-categorical SVM to two-class SVM is *Kesler's Construction* and the other one is *Convergence of the Fixed-Increment Rule*, in both of them as the database increases, speed of the system for SVM training and testing will be slower [7, 18].

Comparison of different classification techniques for music genre classification is explored in [3]. GMM, SVM, KNN and LDA techniques are compared in this study to investigate the use of DWCH feature extraction technique. KNN and LDA are successfully used in many classification tasks, while KNN has been applied in various musical analysis problem. LDA is generally used as a discriminative feature transformation and dimension reduction tool. In [3], it is noted that SVMs are always the best classifiers for content-based music genre classification. It is also observed that classification variations on different classification techniques are small.

HMM is yet another well-known statistical technique to model temporal correlations, and mostly used in speech recognition systems. Most of the classification methods in music genre classification are supervised learning, i.e learner has to generalize from present data to unseen situations. But unsupervised cases, when we do not know the state of the nature for each sample, are also available. Unsupervised classification of music genre using HMM is studied in [9]. In this study, HMM is used to build an unsupervised clustering method based upon the given measure of similarity. In this research area that was the first time for unsupervised clustering to be used for music genre classification. In order to classify music genre generally large number of training samples have been collected for each genre and they are labeled with suitable genre types. This limits the number of genres used in the study. In order to achive this time consuming job, the unsupervised classification technique was proposed. The main approach in this work contains two parts, in first part every individual music file segmented into parts and each part is segmented again into smaller parts on the basis of their content rhythmic structure. After features are extracted from those smaller

parts, HMM is used as a trainer for those features. In the last part of the main approach, every HMM pair are put into a distance matrix and clustering is generated on this matrix. Unlike previously defined features, in this approach the extracted music features are time series data. Hence, HMM is suitable to be used to model this time series data.

Most of the reviewed systems classify music genre types according to the same genre independent feature set in an universal pool of genre types [3, 6]. However, it is clear that different genre types have different classification criteria, thus some features are more suitable than others in separating some genre types from others. For example, the beat strength is likely to perform better in discrimination of classical and pop than of rock and pop music. A hierarchical classification scheme enables us to use different features for different genres. It is also more likely to produce more acceptable misclassifications within higher-level genre.

In [7], music is first classified into two meta-classes (pop/classical or rock/jazz) according to the beat spectrum and LP-derived cepstrum. Later, the pop/classical music is further classified (into pop or classical) according to LP-derived cepstrum, spectrum power, and MFCCs. The rock/jazz music is classified (into rock or jazz) according to zero crossing rate and MFCCs [19].

Hierarchical music genre classification avails easier and flexible music browsing, increases classification efficiency by lowering the number of classes, and decreases the the confusion between music genre types of different hierarchies. Recently, the hierarchical classification is studied and found promising for music genre classification [7, 8, 18]. Similar to the music genre interpretation, the music signals belonging to the same hierarchy contain common characteristics, such as common types of instruments and common rhythmic patterns. The structure of the music genre hierarchy could be human determined for the flexible browsing or automatically determined for maximizing the classification performance. It is expected to extract similar hierarchy structures for both human and machine determined systems, however the structures may differ due to the different objectives, where in the human determined system ease of browsing and in the machine determined system classification performance are favored.

Burred et al. compared the hierarchical classification scheme with direct classification [20]. In hierarchical classification, signals were first classified into speech, music and back-

ground noise. Music was then classified with a three-level structure, first classifying it into classical and non-classical music, and later classifying it into chamber or orchestral music, or into rock, electronic/pop, and jazz/blues. At the third level, music was classified further, e.g. rock into hard rock or soft rock. They extracted a wide selection of different features and used a feature selection algorithm to select the best performing set of features for each particular sub-genre classification. As a result, they achieved very similar recognition accuracies for hierarchical and direct classification schemes [19].

In [18], relationships of dependence between different genres and information about genre contents are obtained via hierarchical taxonomy. They showed that taxonomy improves the classification performance. They also proposed an approach for automatically generating genre taxonomies based on the confusion matrix via linear discriminant projection.

Chapter 3

MUSIC SIGNAL FEATURE EXTRACTION

This chapter presents the music signal features that are employed in the proposed automatic music genre classification system. The main aim of the feature extraction is to represent a signal in a compressed lower dimensional space, so that in this feature representation, significant information about the signal can be captured and it can be used to emphasize certain discriminative characteristics across different kinds of signals, such as different music genre types.

Most of the time, extraction of audio features is performed in the context of some specific application or analysis technique. So for example, a set of features for representing speech might be evaluated in the context of speech compression while a set of features for representing musical content might be evaluated in the context of automatic musical genre classification. The large variety of audio features that have been proposed differ not only in how effective they are but also in their performance requirements. For example, FFT-based features might be more appropriate for a real time application than a computationally intensive but perceptually more accurate features based on a cochlear filterbank.

Widely, feature extraction is performed using some form of time-frequency analysis technique such as the Short Time Fourier Transform (STFT). Time-frequency analysis technique basically represent the energy distribution of the signal in a time-frequency plane and differ in how this plane is sub-divided into regions. The human ear also acts as a time-frequency analyzer decomposing the incoming acoustic pressure wave in the cochlea into different frequency bands [10].

In this thesis, only a selected set of timbral features are used. Since, the amount of training data is always limited, incorporation of additional features may or may not lead to any measurable error reduction. This does not mean that additional features are not important, but in this study our main focus is to design better classifiers for music genre classification. It is expected to see similar performance gains with the proposed classification

structures if a new and better music feature set is employed. On the other hand, although long-term features, such as pitch, articulation rate etc, include important informations about audio signals, it is difficult to extract them in a robust manner.

Short-time analysis over 25ms overlapping audio windows are performed for the extraction of timbral texture features for each 10ms frame. Hamming window of size 25ms is applied to the analysis audio segment to remove edge effects (spectral leakage). Many different features were preliminarily tested, but only a selected set including the most promising ones were used in the final experiments. Mel-Frequency Cepstral Coefficients (MFCC) have been successfully used in many audio classification problems [21], and they proved to be a good choice for our application, as well. The other selected features are Spectral Centroid (SC), Spectral Roll-Off (SR), Spectral Flux (SF) and Average Zero-Crossing Rate (ZCR). The individual music features are described in detail in the following sections.

3.1 Mel-Frequency Cepstral Coefficients

Music perception happens through the interaction of physical objects that produce sound (musical instruments) with our auditory system. Therefore, it has to be treated as a psychoacoustical phenomenon, not a purely physical one. It is known that human perception of the frequency content of sounds uses a logarithmic scale rather than a linear scale. Mel-scale is a widely used and a good approximation of this logarithmic scale to measure natural response of human perceptual system. Mel-scale modeling is based on the experiments with simple tones in which subjects were required to divide given frequency ranges into four perceptually equal intervals or to adjust the frequency of a stimulus tone to be half as high as that of comparison tone. One *mel* is defined as one thousand of the pitch of a 1kHz tone. Mel-scale models the sensitivity of the human ear more closely than a purely linear scale and provides for greater discriminatory capability between audio segments [22, 23]. An approximation between a frequency value f in Hertz and in Mel is defined by:

$$Mel(f) = 2595 \log_{10} \left(1 + \frac{f}{700}\right) \tag{3.1}$$

MFCC features are extracted using the mel-scale. MFCC features include not only the spectral information but also perceptually relevant parts of the auditory spectrum. Figure 3.1 shows the feature extraction process for MFCC. Pre-emphasis is done using a first-order

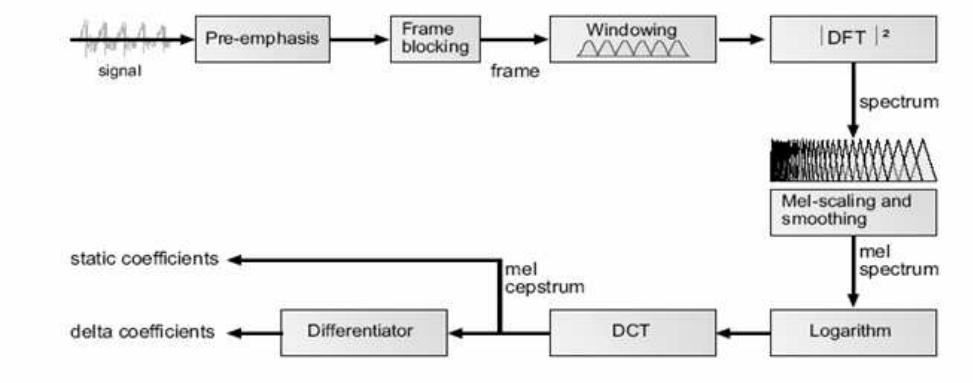

Figure 3.1: MFCC Feature Extraction System Schematic

finite impulse response (FIR) filter $1 - 0.97z^{-1}$ to increase the relative energy of high-frequency spectrum. The aim of frame blocking is to segment the signal into statistically stationary blocks. Hamming window is used to weight the pre-emphasized frames. Next, the discrete Fourier transform (DFT) is calculated for each frames. Since the human auditory system perceives sound better in mel-scale, a perceptually meaningful frequency resolution is obtained by averaging the magnitude spectral components over mel-scaled frequency bins. This is done by using a filterbank consisting of 26 triangular filters, spaced uniformly on the mel-scale. The mel-scale filterbank is shown in Fig. 3.2.

After the mel-scale filterbank, logarithm is applied to the amplitude spectrum, since the perceived loudness of a signal has been found to be approximately logarithmic. The mel-spectral components are highly correlated. This is not a desired property especially for features to be used with Gaussian mixture models. The decorrelation of the mel-spectral components allows the use of diagonal covariance matrices in the subsequent statistical modeling. The mel-spectral components are decorrelated using discrete cosine transform (DCT), which has been empirically found to approximate the Karhunen-Loeve transform. The DCT is calculated by,

$$C_{mel}(i) = \sum_{i=1}^{M} (\log S_j) cos\left(\frac{\pi i}{M}\left(j - \frac{1}{2}\right)\right), \quad i = 1, \ldots, N, \tag{3.2}$$

where $C_{mel}(i)$ is the ith MFCC, $M = 26$ is the total number of channels in filterbank, S_j is the magnitude response of the j th filterbank channel, and N is the total number of the

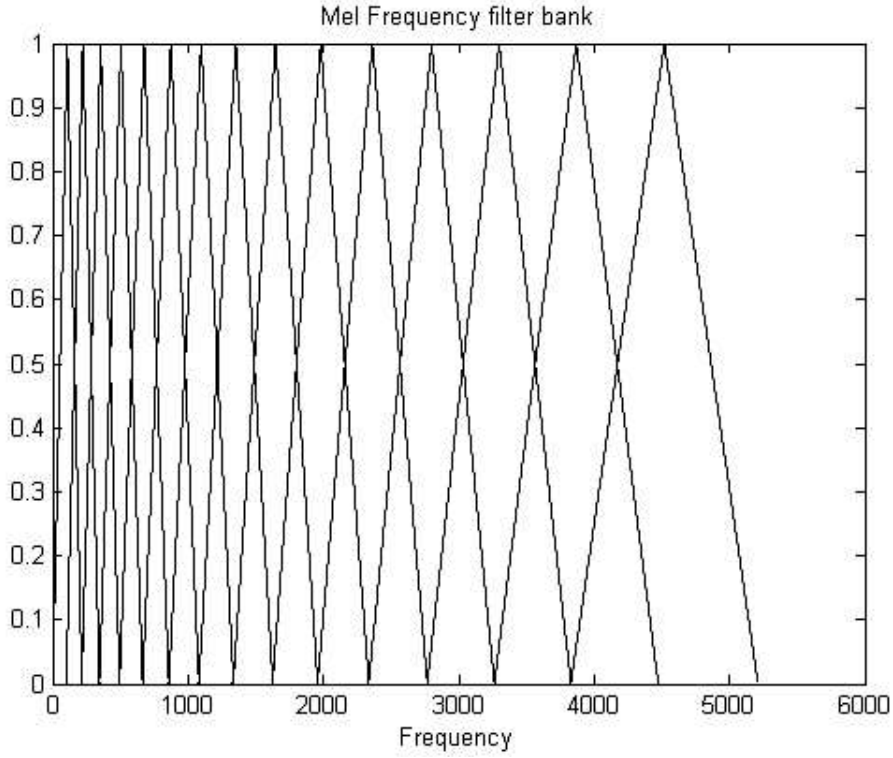

Figure 3.2: Mel-scale filterbank

coefficients. For each frame, $N = 13$ cepstral coefficients are obtained using this transform.

3.2 Spectral Centroid

Centroid can be thought as a balanced point. Center of gravity of the magnitude spectrum of the STFT is defined as Spectral-Centroid. Let $M_t[n]$ is magnitude of the Fourier transform at frame t and frequency bin n, then spectral centroid is:

$$C_t = \frac{\sum_{n=1}^{N} M_t[n] * n}{\sum_{n=1}^{N} M_t[n]}, \tag{3.3}$$

Spectral shape can be understood from centroid information in a way that higher centroid values indicates the "brighter" textures with more high frequencies, and vice versa on the opposite case. Experiments showed that Spectral Centroid characterizates the musical instruments timbre [10].

Many kinds of music involve percussive sounds which, by including high-frequency noise, push the spectral mean higher. In addition, excitation energies can be higher for music than

for speech, where pitch stays in a fairly low range. This measure gives different results for voiced and unvoiced speech [24, 25].

Spectral Centroid not only finds the average frequency with weighted amplitude amplification of spectrum for a given frame, but also finds the nearest spectral bins for that frequency. This process is dependent upon the frame size in practice. For instance when we use FFT size of 1024, bin sizes are nearly $7 - 10$Hz, so the maximum error is not bigger than 5Hz.

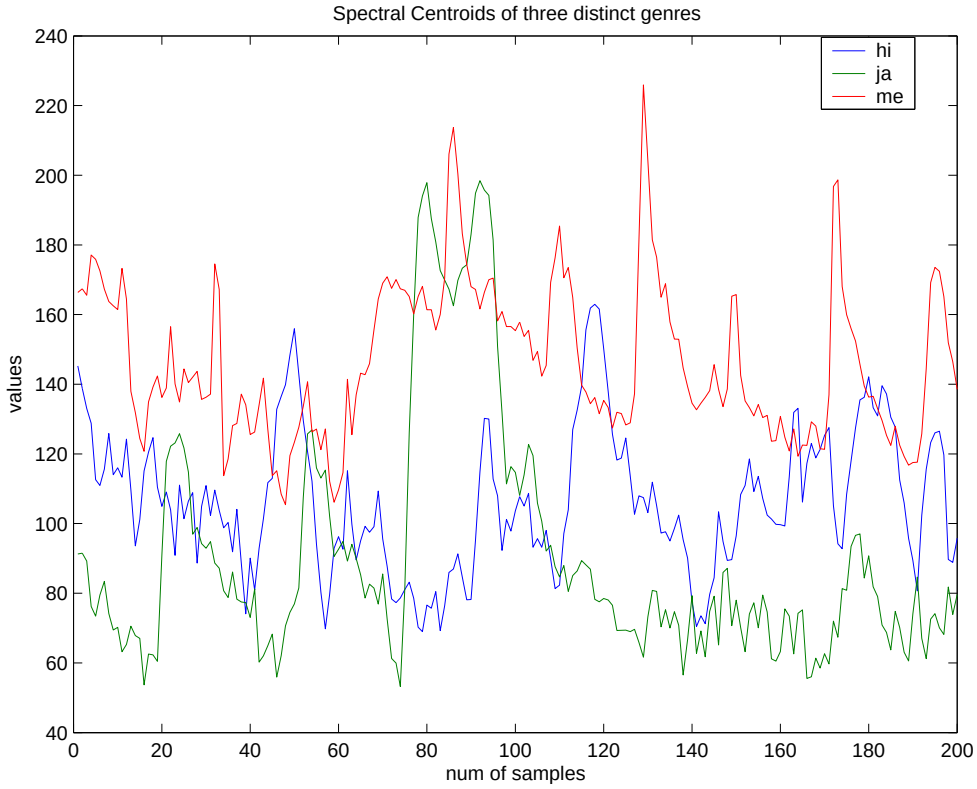

Figure 3.3: A view of Spectral Centroids for hip-hop, jazz and metal genres.

In Fig. 3.3, spectral centroids of three distinct genres are shown. The spectral centroid is observed to be discriminative feature for these genre types. Fig. 3.4 plots the spectral centroid features for the remaining four genre types. In Fig. 3.5, the spectral centroid features of all the genre types are seperately plotted. They observed to have discriminative patterns across different genre types.

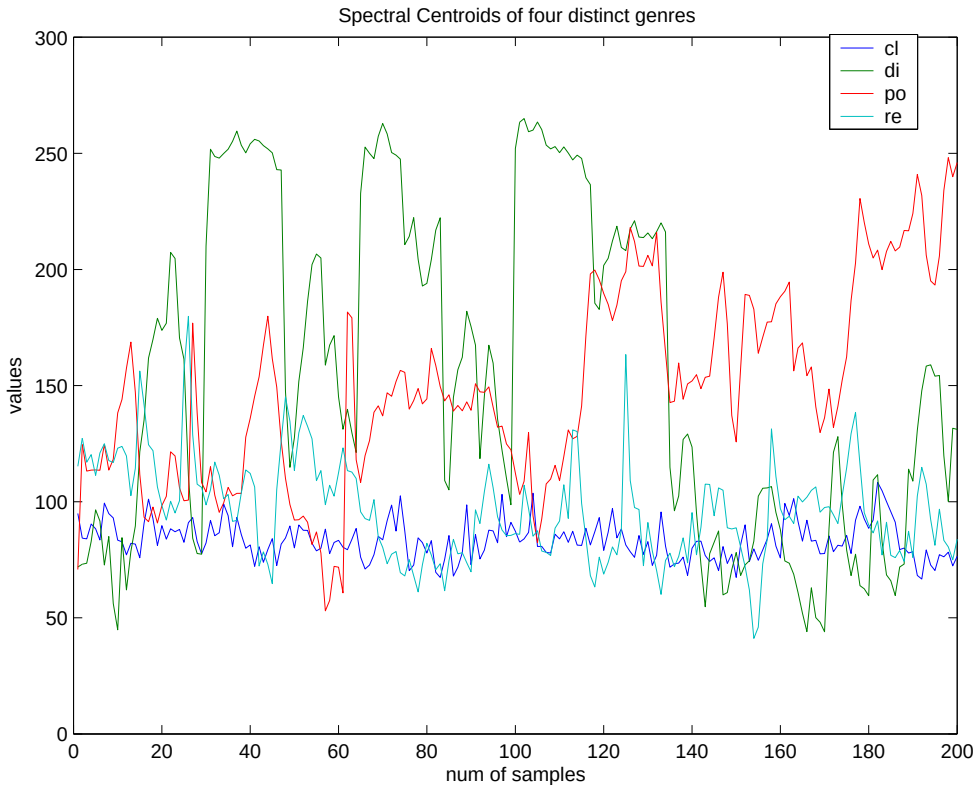

Figure 3.4: A view of spectral centroids of classical, disco, pop and reggae genres

3.3 Spectral Roll-Off

Let R_t be the frequency below which 85% of the magnitude distribution of FFT spectrum is concentrated, then R_t is called the Spectral Roll-Off. The spectral RollOff point R_t determines where 85 % of the window‘s energy is achieved. It is used to distinguish voiced part(music with singer’s voice) from unvoiced part(music without singer’s voice) in speech and music signals. Unvoiced speech has a high proportion of energy contained in the high-frequency range of the spectrum, where most of the energy for voiced speech and music is contained in lower bands. This is a measure of the ”skewness” of the spectral shape. The Spectral Roll-Off can be defined as,

$$\sum_{n=1}^{R_t} M_t[n] = 0.85 * \sum_{n=1}^{N} M_t[n] \tag{3.4}$$

The Spectral Roll-Off is also another measure type of spectral shape like spectral centroid. It shows how much of the energy of the signal is concentrated in the lower frequencies

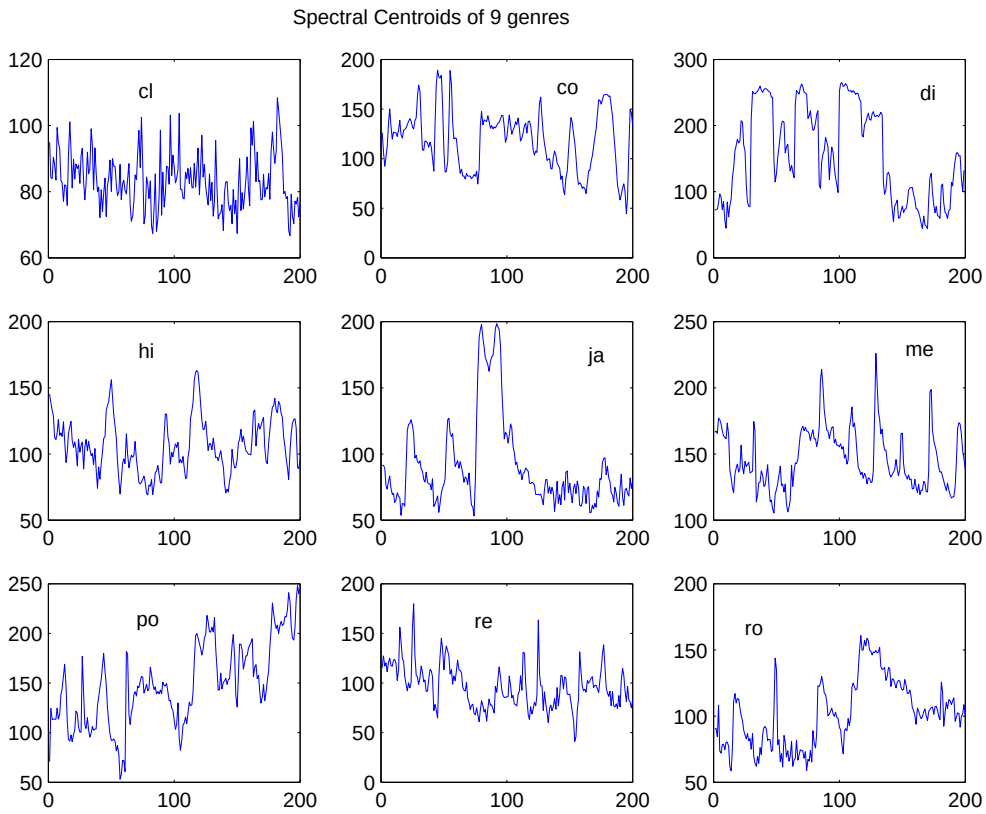

Figure 3.5: A view of spectral centroids of all genres

or on the other way it shows the how much of the signal's energy is concentrated in the higher frequencies. Not all of them but most of the music exhibit a relatively flat average spectrum below 250 Hz and when the Spectral Roll-Off analysis is concerned it is seen that spectral rolloff is nearly in between -3 dB/octave and -9 dB/octave. Rolloff values smaller than -3 dB/octave begin to sound like white noise, and rolloff values heavier than -9 dB/octave sound very dark and dull.

Spectral roll-off plots for various music genre types are given in Figures 3.6, 3.7 and 3.8. As clearly observed from these figures, the spectral roll-off feature is discriminative across different genre types.

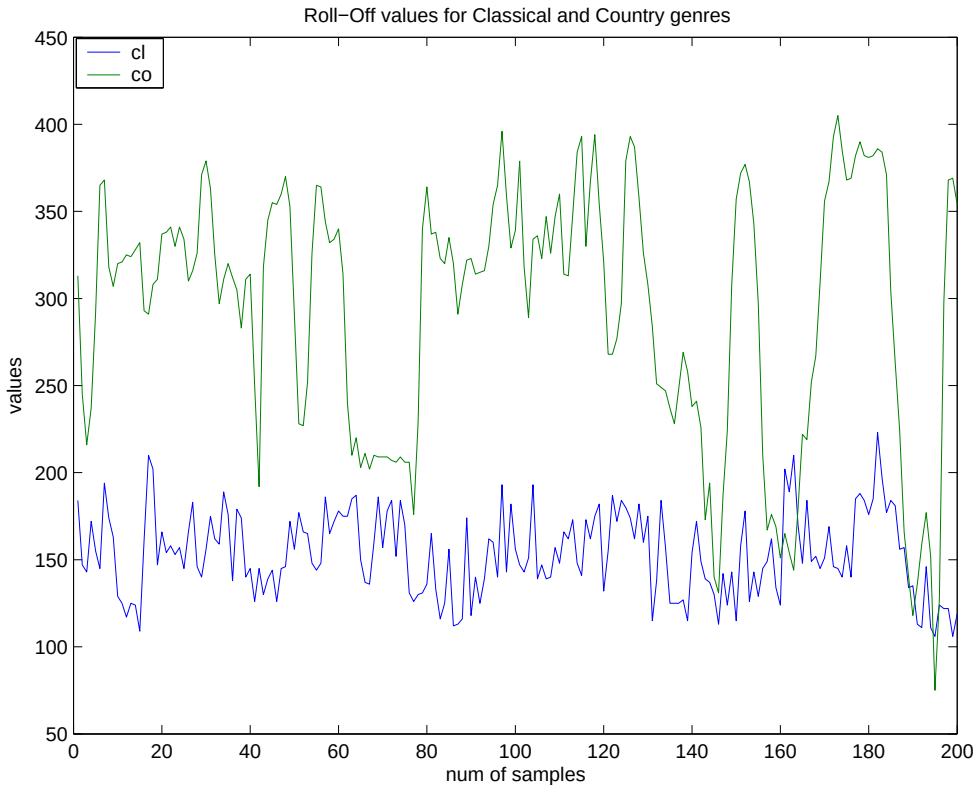

Figure 3.6: A view of spectral roll-Off of classical and country genres

3.4 Spectral Flux

Squared difference between the normalized magnitudes of successive spectral distributions are defined as Spectral Flux. The spectral flux is defined as,

$$F_t = \sum_{n=1}^{N} (N_t[n] - N_{t-1}[n])^2 \tag{3.5}$$

where $N_t[n]$, $N_{t-1}[n]$ are the normalized magnitude of the Fourier transform at the current frame t and the previous frame $t-1$, respectively. The spectral flux is a measure of the amount of local spectral change. Spectral flux has been observed to be an important perceptual attribute in the characterization of musical instrument timbre [10].

Spectral flux, also known as the delta spectrum magnitude, measures the spectral difference between frames, hence it is characterized by the shape of spectrum. Music has a more constant rate with respect to speech signals when spectral flux is considered as a feature. Spectral flux values are higher for voiced speech signals and lower for unvoiced speech and

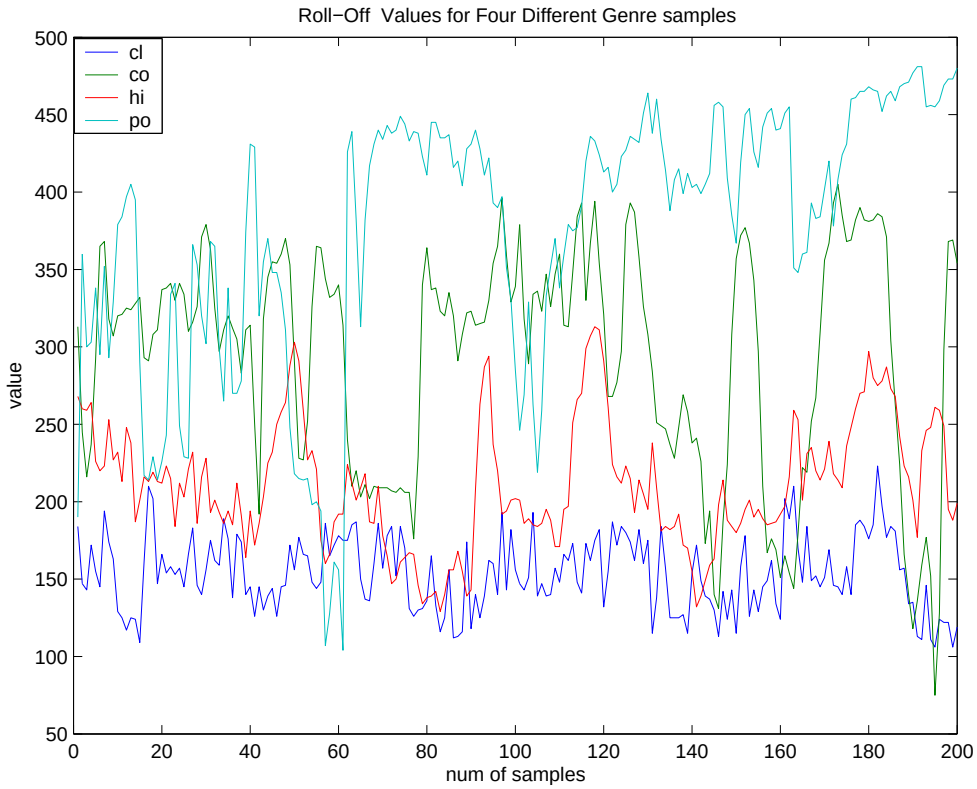

Figure 3.7: A view of spectral roll-off of classical,country, hip-hop and pop genres

music signals. Spectral flux can also be used to separate vocal music pieces from pure instrumental music, for instance most of the classical music is instrumental so it may be used as discriminative feature to separate classical music from other types of music.

Spectral flux values for all genre types are plotted in Fig. 3.9. Pure instrumental music can be separated from vocal music with the spectral flux information. In pure instrumental music, such as in classical music, the spectral flux is observed to have a periodic nature.

3.5 Average Zero-Crossing Rate (ZCR)

A zero-crossing occurs when successive samples in a digital signal have different signs. Therefore, the rate of zero-crossings can be used as a simple measure of a signal's frequency content. For simple signals, the ZCR is directly related to the fundamental frequency (f_0). Since it is a time-domain feature, it can be calculated effectively. The ZCR can been used as a statistical measure of spectral characteristics of music signal. The ZCR also makes it

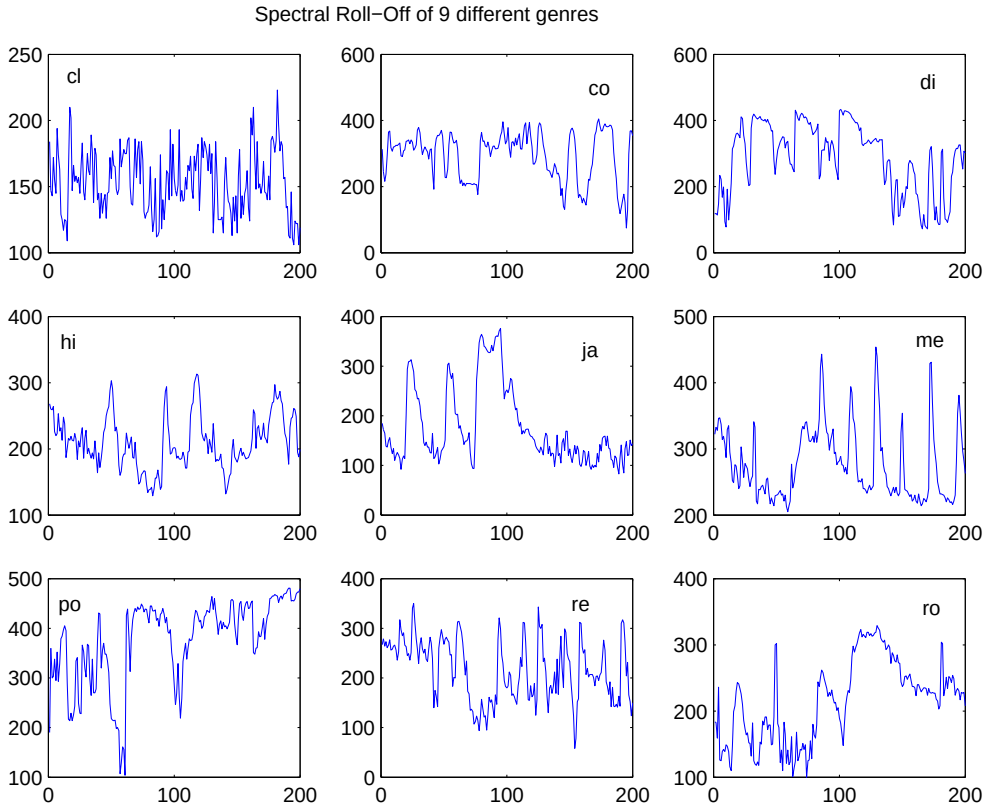

Figure 3.8: A view of spectral roll-off of all genres

possible to differentiate between voiced and unvoiced speech components, voiced components have much higher ZCR values than unvoiced ones. The average short-time zero-crossing rate can also be useful in combination with other features in general audio signal classification systems. An extension of the time domain zero-crossing rate of audio segment is calculated as:

$$Z_{C_m} = \frac{1}{8}\sum_{n} |(sgn(x[n-1]x[n])+1)*(sgn(x[n]x[n+1])-1)*(sgn(x[n+1]x[n+2])+1)| \quad (3.6)$$

where

$$sgn[x(n)] = \begin{cases} 1 & \text{if } x(n) \geq 0 \\ -1 & \text{if } x(n) < 0 \end{cases} \quad (3.7)$$

and index n runs in the m-th audio window.

Time domain ZCR provides a measure of the noisiness of the signal. For example heavy metal music due to guitar distortion and lots of drums tends to have much higher zero

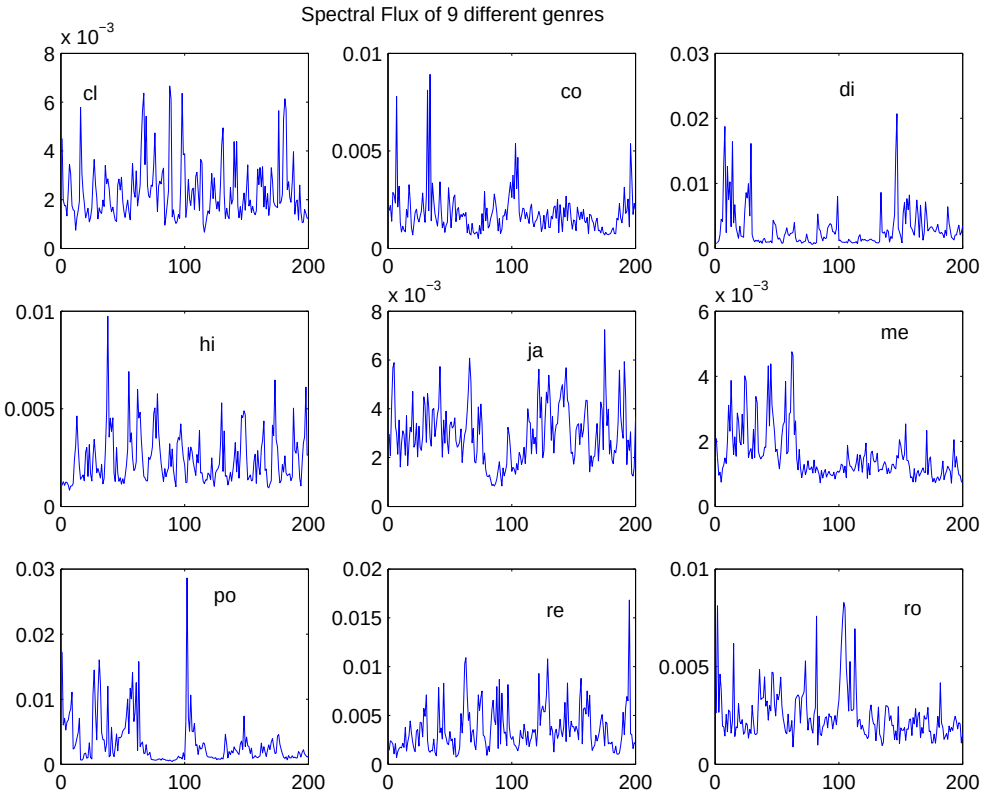

Figure 3.9: A view of spectral flux of all genres

crossing values than classical music. For clean (without noise) signals Zero Crossings are highly correlated with the Spectral Centroid [10].

Statistical measure of spectral characteristics in different music genres are determined by analyzing the changes of ZCR over time in Figures 3.10 and 3.11.

The resulting timbral features, spectral roll-off, spectral centroid, spectral flux, zero crossing rate and 13 dimensional MFCC feature, are combined in a 17 dimensional feature vector, which is plotted in Fig. 3.12.

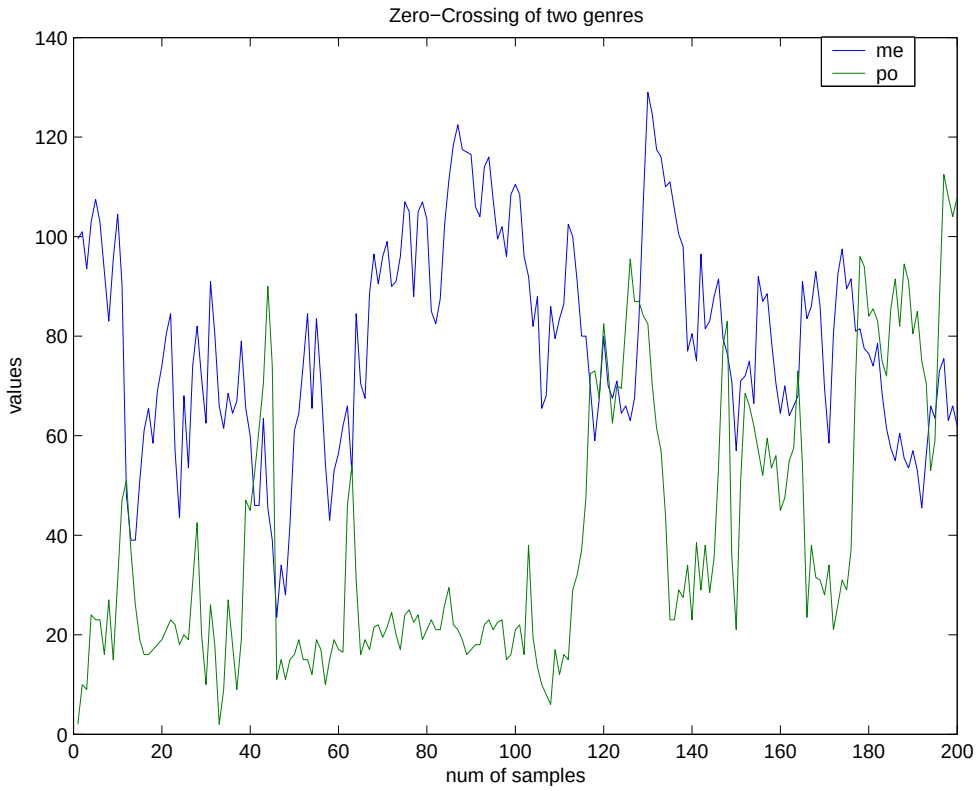

Figure 3.10: A view of zero-crossing of metal and pop genres

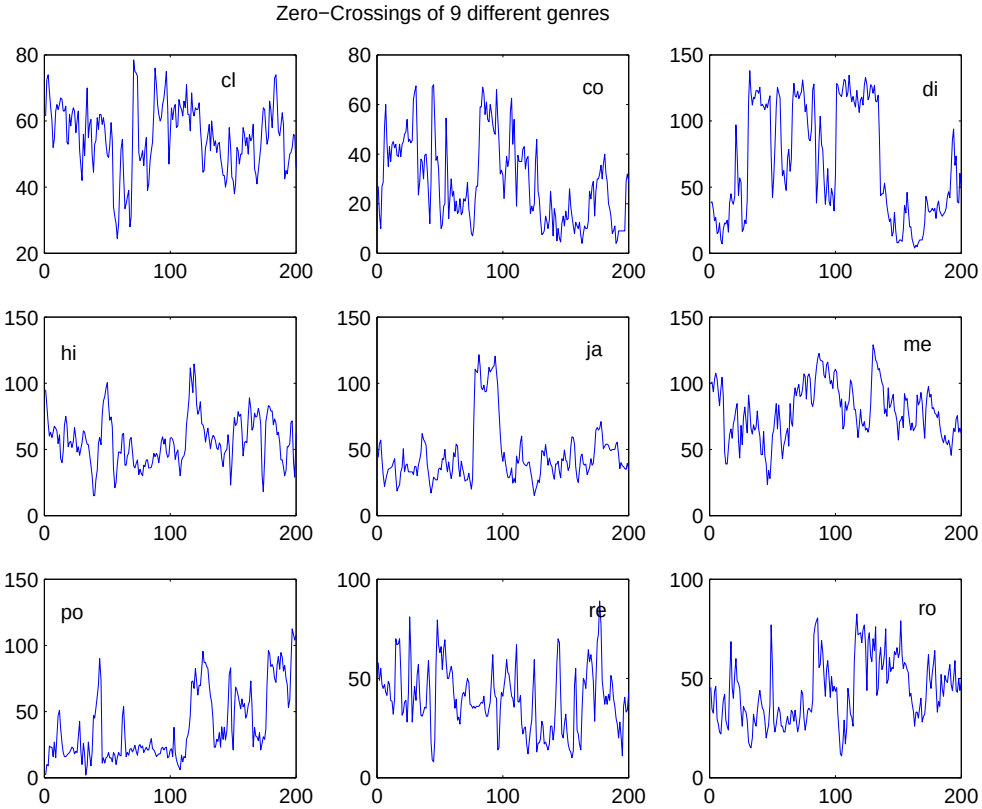

Figure 3.11: A view of zero-crossing of all genres

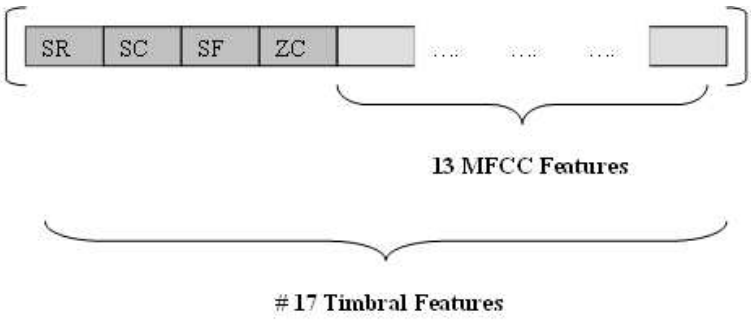

Figure 3.12: Feature vector extracted from a frame demonstration

Chapter 4

MUSIC GENRE CLASSIFICATION

4.1 Classification Overview

Classification can be defined as categorization of patterns into classes(or groups) with respect to patterns. The task of a classification system is to use the feature vector provided by the feature extractor to assign the object to a category [17].

The classifiers are not perfect due to complexity of the sample space. The reason is not because of the models created for the specific problem but it is because of both variability in the feature values for samples(or patterns) in the same class, called intra-class stochastic behavior, and their relative difference of feature values for samples in different classes, called inter-class confusion. There are many ways to decrease imperfectness of the classifiers, and there are many measures of this imperfectness. The simplest one is classification error rate tracking. It is commonly desired to seek minimum error-rate-classification [17]. Structural risk minimization and cost minimization are other two measures of classifiers. To fit a model to the past data to calculate the risk for a new application and then decides to accept or refuse it accordingly is the main idea in classification, in which risk function can be chosen with respect to types of problem[26].

Two different types of classification approaches have been used in the automatic musical genre recognition systems. The most common approach is to classify each frame separately, and to combine these classification results over an analysis segment in order to get the global classification result. The second approach is to also take into account the temporal relationships between frames in the classifier. For humans, order of frames is an important property. For instance, the recognition accuracy has been observed to decrease from 85% to 70% when the order of frames was mixed [14].

There are many kinds of classification methods proposed in the literature and boundaries between these classification methods cannot be defined strictly on the light of which is the best classifiers. It actually depends on the some prospects such as: training time, amount

of training data required, classification time, robustness, and generalization.

The main idea behind most of these classification schemes is to "learn" the geometric structure of the training data in the feature space and use that to estimate a "model" that can generalize to new unclassified patterns. Parametric classifiers assume a functional form for the underlying feature probability distribution while non-parametric ones use directly the training set for classification [10]. More details can be found in Duda's textbook on Pattern Classification [17].

4.1.1 The State-Of-Art

An overall view of our system is presented in Fig. 4.1. In the general block-diagram level, all modern pattern recognition systems share the same basic structure. The system consists of four parts: preprocessing, feature extraction, pattern learning, and classification. At the preprocessing stage, the input signal is normalized before extracting features, which are used to characterize the signal. In order to train the recognition system, we need to have a sufficient amount of training samples from each class to be recognized. In the pattern learning stage, a representative pattern will be determined for the features of the actual class. In the classification stage, the test data is compared to previously calculated models and classification is done by measuring the similarity between the test data and each model, and assigning the unknown observation to the class whose model is most similar to the observation [19].

In this study, class-conditional probability density functions, $p(f|\lambda)$, are estimated in the pattern learning phase, where f and λ are respectively the feature vector and the genre class model. Let $\lambda_1, \lambda_2, \ldots, \lambda_N$ be the N different genre models in the database. In the music genre classification process, given a sequence of features, $\{f_1, f_2, \ldots, f_K\}$, which are extracted from a decision window of music signal, one can find the most likely music genre class, λ^*, by maximizing the joint class-conditional probability,

$$\lambda^* = \arg\max_{\lambda_n} \sum_{k=1}^{K} \log p(f_k|\lambda_n). \tag{4.1}$$

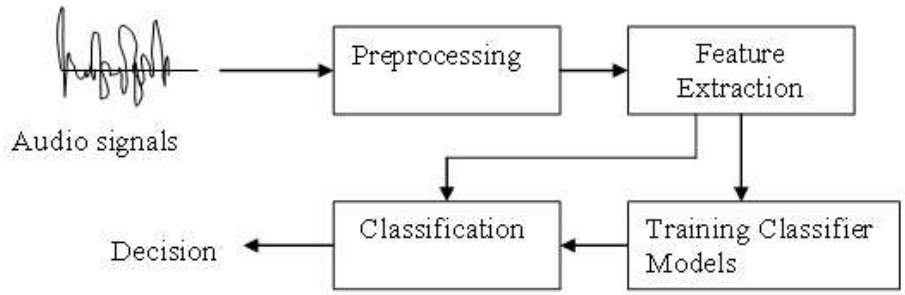

Figure 4.1: General structure of classification system

4.2 Gaussian Mixture Model Classifiers

This chapter presents the theoretical framework of widely used audio classification modeling technique, GMM. GMM is used to estimate the probability density of each genre class over the feature space. The probability density function is defined as the weighted sum of Gaussian densities, which are called mixtures. One widely used technique to model variability in audio signals is to use multi-dimensional Gaussian pdf in which temporal ordering of feature vectors are not concerned.

Although various density functions that have been investigated, none has received more attention than the multivariate gaussian density. To a large extent, this attention is due to its analytic tractability. However, multivariate normal density is also an appropriate model for an important situation, namely, the case where the feature vectors x for a given class w_i are continuous-valued, randomly corrupted versions of a single typical or prototype vector μ_i. There is deep relationship between the normal distribution and entropy, it can be shown that the normal distribution has the maximum entropy of all distribution having a given mean and variance [17].

Gaussian mixture density modeling has been widely used in various disciplines that require signal characterization for classification and recognition, as well as estimation of unknown probability densities. In this thesis, the Gaussian Mixture Models (GMM) are

used for the class-conditional probability density function estimation.

4.2.1 Mathematical Model of GMM

The general d-dimensional normal density is written as

$$p(\mathbf{x}) = \frac{1}{(2\pi)^{d/2}|\mathbf{\Sigma}|^{1/2}} exp[-\frac{1}{2}(\mathbf{x}-\mu)^t \mathbf{\Sigma}^{-1}(\mathbf{x}-\mu)], \tag{4.2}$$

where $\mathbf{x}$ is a d-dimensional feature vector, μ is d-dimensional mean vector, $\mathbf{\Sigma}$ is the d-by-d covariance matrix and finally $|\mathbf{\Sigma}|$ and $\mathbf{\Sigma}^{-1}$ are the determinant and inverse of covariance matrix respectively. Eq. 4.2 is represented as,

$$p(\mathbf{x}) \sim N(\mu, \mathbf{\Sigma}). \tag{4.3}$$

In equations 4.2 and 4.3, μ is the mean vector of feature vectors, in which each of the feature vector is represented as μ_i and it is in d-dimension. Similarly $\mathbf{\Sigma}$ is the covariance matrix of feature vectors and σ_{ij} is the ijth component of $\mathbf{\Sigma}$.

If the data is normally distributed and μ and Σ are known, the conditional density function $p(x|\lambda_i)$ in the classifier discriminant function can be replaced with equation 4.2. In general, we have to estimate the parameters that rely on the general form of the distribution.

In most classification problems, the conditional densities are not known. However, in many cases, a reasonable assumption can be made about their general form. This makes the problem significantly easier, since we need only estimate the parameters of the functions, not the functions themselves.

The unknown probability densities are usually estimated in a training process, using sample data. For instance, it might be assumed that $p(x|\lambda_i)$ is a normal density. We then need to find the values of the mean μ and the covariance Σ.

The probability of a feature vector being in any one of N classes, the classes are "genres" in our case, for a particular music genre model, denoted by λ, is represented by the union, or *mixtures*, of different gaussian pdfs:

$$p(\mathbf{x}|\lambda) = \sum_{i=1}^{N} p_i b_i(\mathbf{x}), \tag{4.4}$$

where $b_i(\mathbf{x})$ are the component mixture densities,

$$b_i(\mathbf{x}) = \frac{1}{(2\pi)^{d/2}|\mathbf{\Sigma}_i|^{1/2}} exp[-\frac{1}{2}(\mathbf{x}-\mu_i)^t \mathbf{\Sigma}_i^{-1}(\mathbf{x}-\mu_i)], \tag{4.5}$$

and p_i are the mixture weights. Given that each individual Gaussian pdf integrates to unity, then we constrain $\sum_{i=1}^{N} p_i = 1$ to ensure that the mixture density represent a true pdf. The music genre model λ then represent the set of GMM mean, covariance and weight parameters, i.e.,

$$\lambda = (p_i, \mu_i, \mathbf{\Sigma}_i). \tag{4.6}$$

GMM is a "soft" representation of the various classes that make up the sound of the each genre, in which each component density can be thought as distribution of possible feature vectors associated with each of the N classes. We have to estimate (train) parameters of GMM for each genre. Like in the VQ, within the training data, clusters should be formed and in further step, different than the VQ we have to represent each cluster with a Gaussian pdf, the union of Gaussian pdfs being the GMM. Namely, we can say that GMM is "VQ on steroids", each cluster has not only mean but also associated covariance matrix.

One way to estimate the GMM model parameters is by maximum-likelihood estimation, that is maximizing, with respect to λ, the conditional probability $p(X|\lambda)$ where $X = \{x_0, x_1, \ldots, x_{M-1}\}$ is the collection of all feature vectors for a particular musical genre. A significant property of maximum-likelihood estimation is that for a large enough set of training feature vectors, the model estimate converges to the true model parameters; however, there is no closed-form solution for the GMM representation, thus iterative way is the most common approach to solve the problem.

This iterative solution is performed using the *expectation-maximization* **(EM)** algorithm. The EM algorithm iteratively improves on the GMM parameter estimates by increasing on each iteration the probability that the model estimate λ matches the observed feature vectors. **EM** algorithm is similar to K-means algorithm except that in **EM** relative importance of each component in pdf is concerned. (For details see [23]). The **EM** is briefly described in the following.

EM Algorithm

- **Start** with data set X of N feature vectors x_n, $n = 1, \ldots, N$, initial set of K Gaussian component pdfs $N_k \sim N(\mu_k, \mathbf{\Sigma}_k)$ and K mixture weights $P(k), k = 1, \ldots, K$

- **E-Step:** Determine responsibility $P(k|\mathbf{x}_n)$ of each component pdf N_k for each training data point x_n as

$$p_{kn} \equiv P(k|\mathbf{x}_n) = \frac{p(\mathbf{x}_n|k)P(k)}{p(\mathbf{x}_n)}, \tag{4.7}$$

  with GMM likelihood $p(\mathbf{x}_n) = \sum_{k=1}^{K} p(\mathbf{x}_n|k)P(k)$.

- **M-Step:** Re-estimate component pdfs and weights, based on data and responsibilities:

$$\hat{P(k)} = \frac{1}{N}\sum_{n=1}^{N} p_{kn}, \qquad \hat{\mu}_k = \frac{\sum_n p_{kn}\mathbf{x}_n}{\sum_n p_{kn}}, \tag{4.8}$$

  and $\mathbf{\Sigma}$ depends upon the type of the covariance matrix, full covariance, diagonal covariance and spherical covariance matrix. For each of covariance matrix re-estimation process is also repeated for $\mathbf{\Sigma}$, that is:

$$\hat{\mathbf{\Sigma}}_k = \frac{\sum_n p_{kn}(\mathbf{x}_n - \hat{\mu}_k)(\mathbf{x}_n - \hat{\mu}_k)^T}{\sum_n p_{kn}} \qquad (full.cov), \tag{4.9}$$

$$\hat{\sigma}_{ik} = \frac{\sum_n p_{kn}(x_{in} - \hat{\mu}_{ik})^2}{\sum_n p_{kn}}, i = 1, \ldots, D \quad (diag.cov) \tag{4.10}$$

  and finally,

$$\hat{\sigma}_k^2 = \frac{\sum_n p_{kn}\|\mathbf{x}_n - \hat{\mu}_\mathbf{k}\|^2}{D\sum_n p_{kn}} \quad (spherical.cov). \tag{4.11}$$

- Repeats E-step and M-step until GMM likelihood $p(X) = \prod_{n=1}^{N} p(\mathbf{x}_n)$ of entire data set does not change appreciably, or limit on the number of iterations is reached (For details see [27]).

EM algorithm increases the likelihood function of data iteratively and converges to a stationary parameter vector by letting likelihood function increase monotonically. However, we have to pay attention to initial values of the problem like other optimization algorithms depending on the initial settings. Hence, initial conditions(placement) of Gaussian components are principal significance for convergence of **EM** algorithm. In this case, K-means algorithm is used to indicate initial points of **EM** algorithm.

4.3 Boosting

Boosting is a method of training many simple classifiers, and combining them into a single strong classifier [28]. The main aim of boosting concept is to increase the accuracy of any given learning algorithm. Firstly, we should create a classifier with having accuracy on the training part greater than the average. That is the first restriction to the boosting algorithm, its cause will be presented next. Performance of boosting algorithm also depends on factors such as type of data, number of boosting iteration,noisiness of data, capacity of weak leaner and margin distribution over training examples. After creating classifier having the training rate of above average, new components classifiers have to be added to form an ensemble whose joint decision rule has arbitrarily high accuracy on training set. In such situation it is called that the classification performance is *"boosted"* [17].

The main approach is to generate a sequence of weak classifiers on a weighted set of training samples, and to adapt the training sample weights such that in each new classifier the hard-to-classify samples are weighted more than easy-to-classify samples. The outputs of the weak classifiers are then combined to guarantee certain bounds on error rate [29, 28]. Many variations of the boosting algorithms exist, and they are used successfully in a wide range of problems. In this thesis, we used the first multi-class extension of widely used **AdaBoost** algorithm [29].

There are many different algorithms on basic boosting method as mentioned earlier. The most popular one is the Ada-Boost coming from *Adaptive boosting*, which allows users to add weak learners until some desired low training error has been achieved. In AdaBoost, each training pattern receives a weight that determines its probability of being selected for a training set. When a training pattern is classified correctly, then its chance of being used again in subsequent component classifier is reduced; with similar concept if a training pattern is classified wrongly, then its chance of being used again is increased.

Boosting improves the classification only if the classifier's performance is better than random guess, however, this is not guaranteed a priori. Boosting often does not overfit. *Margin theory* explains this lack of overfitting to some extent. According to margin theory, if the margin of a classifier is large, then it is tended to perform well on test data, if the margin of a classifier is small, then it is not tended to perform well.

Considering GMM classifiers, the maximum-likelihood method has been one of the most

commonly used technique to estimate the parameters of the mixture densities [30]. In this thesis, we propose a modification to the expectation-maximization (*EM*) based training of mixture densities to adapt the weighting approach of the boosting algorithm.

The following subsections describe the first extension of AdaBoost, **AdaBoost.M1**, and the modified *EM* algorithm for boosting GMMs.

4.3.1 Theory of Boosting-AdaBoost Algorithm

Let us have an input sequence of N samples $\{(x_1, y_1), \ldots, (x_N, y_N)\}$ with class labels $y_i \in \mathbf{Y} = \{1, \ldots, k\}$.

Initialize the sample weight vector uniformly:

$$\omega_i^1 = 1/N \quad \text{for} \quad i = 1, \ldots, N.$$

Do for $t = 1, 2, \ldots, T$

i. Train the weak classifier providing it with the sample weights $\mathbf{w}^t$; get back a hypothesis $h_t : \mathbf{X} \to \mathbf{Y}$

ii. Calculate the error of $h_t : \epsilon_t = \sum_{i=1}^{N} \omega_i^t \langle h_t(x_i) \neq y_i \rangle$. If $\epsilon_t > 1/2$, then set $T = t - 1$ and abort loop.

iii. Set $\beta_t = \epsilon_t / (1 - \epsilon_t)$.

iv. Set the new weight vector to be $\omega_i^{t+1} = \omega_i^t \beta_t^{1 - \langle h_t(x_i) \neq y_i \rangle}$; normalize $\mathbf{w}^{t+1}$ such that $\sum_{i=1}^{N} \omega_i^{t+1} = 1$.

Final hypothesis is given as,

$$h_f(x) = \arg\max_{y \in \mathbf{Y}} \sum_{t=1}^{T} \left(\log \frac{1}{\beta_t} \right) \langle h_t(x) = y \rangle. \tag{4.12}$$

In the **AdaBoost.M1** algorithm T specifies the number of iterations in the boosting, and for any predicate π, $\langle \pi \rangle$ is defined to be 1 if π holds and 0 otherwise.

4.3.2 Modified EM *Algorithm*

Let us have an input sequence of N samples $\{x_1, \ldots, x_N\}$ with the associated sample weights $\{\omega_1, \ldots, \omega_N\}$. The underlying density function for K mixtures is given as,

$$f(x) = \sum_{k=1}^{K} P(k)\mathcal{N}_k(\mu_k, \Sigma_k), \tag{4.13}$$

where $P(k)$ values are the mixture weights, μ_k and Σ_k are the respectively mean vector and covariance matrix of the k-th Gaussian mixture density $\mathcal{N}_k$. The modified *EM* algorithm to train the mixture densities over our sample data is given as:

Initialize mixture weights, means and variances. Repeat the following E-step and M-step until achieving convergence.

E-step: Calculate the weighted responsibility $p(k|x_n)$ of each Gaussian mixture $\mathcal{N}_k$ for each training data point x_n as,

$$p_{kn} = \omega_n p(k|x_n) = \omega_n \frac{p(x_n|k)P(k)}{p(x_n)}, \tag{4.14}$$

where $p(x_n)$ can be calculated as, $p(x_n) = \sum_{k=1}^{K} p(x_n|k)P(k)$.

M-step: Re-estimate mixture densities and weights, based on data and weighted responsibilities,

$$\hat{P}(k) = \frac{\sum_{n=1}^{N} p_{kn}}{\sum_k \sum_n p_{kn}}, \quad \hat{\mu}_k = \frac{\sum_{n=1}^{N} p_{kn} x_n}{\sum_n p_{kn}}, \tag{4.15}$$

and further a diagonal covariance matrix is estimated as,

$$\hat{\sigma}_{ik}^2 = \frac{\sum_{n=1}^{N} p_{kn}(x_n - \hat{\mu}_k)(x_n - \hat{\mu}_k)^T}{\sum_n p_{kn}}. \tag{4.16}$$

Note that, the weighted responsibilities de-emphasize the easy-to-classify samples, hence hard-to-classify samples are better modeled by the resulting Gaussian mixture density function.

4.4 Boosting with Inter-Genre Similarity Information

The music signals belonging to the same genre contain certain common characteristics since they are composed of similar types of instruments with similar rhythmic patterns. These common characteristics are captured with statistical pattern recognition methods to achieve the automatic music genre classification [3, 4, 5, 6]. The music genre classification problem

is challenging, especially if the decision window spans a short duration, such as a couple of seconds. One can also expect to observe similarities of spectral content and rhythmic patterns across different music genre types, and with a short decision window mis-classification and confusion rates increase. Three novel approaches are proposed to build discriminative musical genre classifiers that are aware of the similarities across genre models. The first approach addresses capturing of the inter-genre similarities to decrease the level of confusion across similar music genre types. The inter-genre similarity modeling is closely related with the boosting idea, where in boosting the main motivation is to better model the hard-to-classify samples and in inter-genre similarity modeling the main motivation is to cluster the hard-to-classify samples to form the inter-genre similarity. The later approach presents an automatic clustering scheme to determine similar music genre types in a hierarchical classifier architecture.

4.4.1 Inter-Genre Similarity Modeling (IGS)

The timbral texture features represent the short-term spectral content of music signals. Since, music signals may include similar instruments and similar rhythmic patterns, no sharp boundaries between certain different genre types exist. The inter-genre similarity modeling (IGS) is proposed to capture the similar spectral contents among different genre types. Once the IGS clusters are statistically modeled, the IGS frames can be captured and removed from the decision process to reduce the inter-genre confusion. The main idea is to capture most similar frames in each genre, those are classified wrongly when train data is used as test data. All the frames, which are classified wrongly when training data is tested with itself, are combined together and new model called IGS is created from these combined frames.

Let $\lambda_1, \lambda_2, \ldots, \lambda_N$ be the N different genre models in the database. In this study, the Gaussian Mixture Models (GMM) are used for the class-conditional probability density function estimation, $p(f|\lambda)$, where f and λ are respectively the feature vector and the genre class model. The construction of the IGS clusters and the class-conditional statistical modeling can be achieved with the following steps:

i. Perform the statistical modeling of each genre in the database using the available training data of the corresponding music genre class.

ii. Perform frame based genre identification task over the training data, and label each frame as a true-classification or mis-classification.

iii. Construct the statistical model, λ_{IGS}, for the IGS cluster over all the mis-classified frames among all the music genre types.

iv. Update all the N-class music genre models, λ_n, over the true-classified frames.

The above construction creates N-class music genre models and a single-class IGS model. In the music genre identification process, given a sequence of features, $\{f_1, f_2, \ldots, f_K\}$, which are extracted from a window of music signal, one can find the most likely music genre class, λ^*, by maximizing the weighted joint class-conditional probability,

$$\lambda^* = \arg\max_{\lambda_n} \frac{1}{\sum_k \omega_{kn}} \sum_{k=1}^{K} \omega_{kn} \log p(f_k|\lambda_n) \tag{4.17}$$

where the weights ω_{kn} are defined based on the class-conditional IGS model as following,

$$\omega_{kn} = \begin{cases} 1 & \text{if } p(f_k|\lambda_n) > p(f_k|\lambda_{IGS}) \\ 0 & \text{else} \end{cases} \tag{4.18}$$

The proposed weighted joint class-conditional probability maximization eliminates the IGS frames for each music genre from the decision process. The inter-genre confusion decreases and the genre classification rate increases with the resulting discriminative decision process. Experimental results, which are supporting the discrimination based on the IGS elimination, are presented in Sec 5. In Fig. 4.4.1, the dashed area plots the mis-classified frames among training frames, the remaining part includes the frames classified correctly over the training data.

4.5 Hierarchical Classification with Auto-Clustering

Hierarchical classification with taxonomy is a new research area in music genre classification, in hierarchical classification there has been little work done so far. Hierarchical music genre classification avails easier and flexible music browsing, increases classification efficiency by lowering the number of classes, and decreases the confusion between music genre types of different hierarchies. Recently, the hierarchical classification is studied and found promising for music genre classification [7, 18].

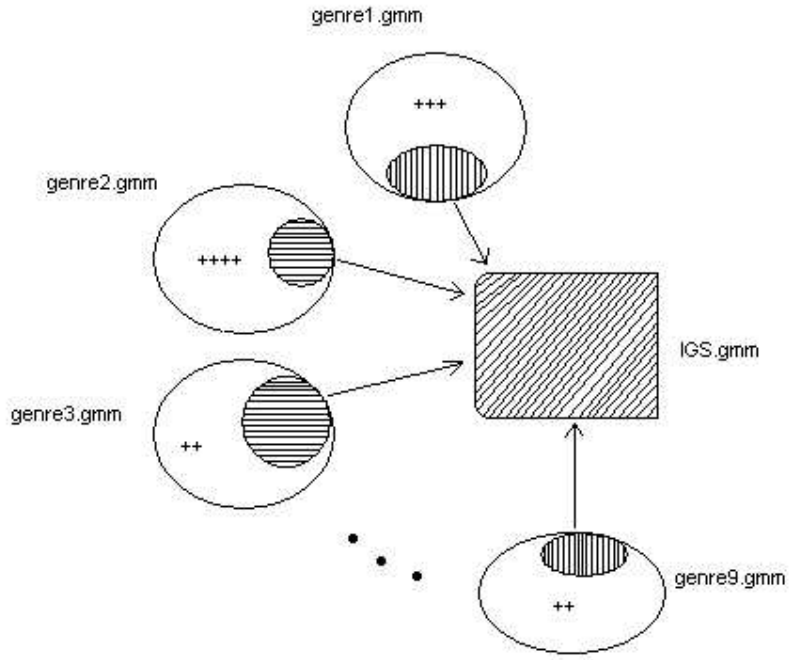

Figure 4.2: Capturing most-confusing frames into IGS

All musical pieces are similar but some are more similar than others, on the other way, it can be said that some genres are similar to some other genres, similar to the music genre interpretation, since the music signals belonging to the same hierarchy contain common characteristics, such as common types of instruments and common rhythmic patterns. These similarities are the basis of hierarchical clustering in music genre classification. The structure of the music genre hierarchy could be human determined for the flexible browsing or automatically determined for maximizing the classification performance. It is expected to extract similar hierarchy structures for both human and machine determined systems, however the structures may differ due to the different objectives, where in the human determined system ease of browsing and in the machine determined system classification performance are favored.

In this section, a new auto-clustering algorithm for the hierarchical structure extraction is proposed. The auto-clustering algorithm is iterative and depends on the confusion matrix,

where it generates an hierarchy tree by combining the most confused two clusters in each iteration. Let $C = \{c_{ij}\}$ be the $N \times N$ confusion matrix, which is calculated over the training data, for N-class music genre classification, and c_{ij} is the confusion rate of i-th music genre to the j-th one. Then, the auto-clustering algorithm is defined as,

i. Find the most confused two genre clusters i^* and j^*,

$$(i^*, j^*) = \arg \max_{ij \text{ s.t. } i \neq j} c_{ij}.$$

ii. Combine the genre clusters i^* and j^* into a new genre cluster, estimate the class-conditional pdf models for each cluster using the training data, and compute the confusion matrix of the new $(N-1)$-class structure using the same training data.

iii. Repeat steps (i) and (ii) to further reduce the number of clusters to M in the hierarchical structure, where $M < (N-1)$.

The classification task starts with an M-class identification system, then the identification process is repeated within the identified cluster until reaching the lowest level in the hierarchy.

In the experimental studies, hierarchical music genre classification is tested with human and machine determined hierarchy structures, and also with the integration of discriminative IGS clustering.

Chapter 5

RESULTS AND DISCUSSION

A proper database should be used to evaluate a classification system, in which choosing the data in an appropriate way is as important as modeling the system. We have two different data sets including 9 different genre types. Country, classical, disco, hiphop, jazz, metal, pop, reggae and rock are the 9 distinct music genre types in our system. The first database, DA, is taken from Tzanetakis [10]. In the DA database each genre includes 20 different representative audio segments of duration 30s for each of 9 music genre types, resulting a total duration of $9x20x30 = 5400$ seconds. In the second database, DB, each genre includes 35 different representative audio segments of duration 30s for each 9 music genre types. In both databases, all the audio files are stored mono at 22050Hz with 16-bit words.

In DA, the training and testing schemes are employed in two different scenarios. In the first scenario (TTA), the training and testing are repeated for 10 independent data partitions and the average performances are reported. Each data partition includes respectively 18 and 2 audio segments for each genre type in training and testing. In the second scenario (TTB), the database is split into two partitions, each partition includes 10 different audio segments from each genre type, where they are used in alternating order for training and testing of the music genre classifiers. In DB, since each genre includes 35 different audio signals, database is split into two groups of 15 and 20 audio segments, where they are used in alternating order for training and testing.

The main difference between TTA and TTB scenarios is that TTA scenario has significantly more training data compared to TTB scenario that improves classification performance for TTA. The average performances over these training and testing scenarios are reported in the following sections.

The average correct classification rates of the two testing scenarios over all music genre types for varying number of Gaussian mixtures are given in Table 5.1. Each classification

Table 5.1: Average identification rates (%) for varying number of Gaussian mixture densities over three different decision window sizes (one audio frame (10ms), 1s and 3s windows).

Average Identification Rates (%)					
Classifier	TTA			TTB	
	10 ms	1 s	3 s	1 s	3 s
GMM(8)	34.85	61.90	65.15	55.56	57.45
GMM(16)	36.89	65.76	68.71	57.90	60.12
GMM(32)	38.46	67.53	70.13	57.69	59.25
GMM(64)	39.75	69.54	72.64	-	-
GMM(128)	39.95	69.69	71.85	-	-

decision is given for a decision window whose size is picked to be 10ms, 1s or 3s. The class conditional probability of a decision window is calculated by multiplying all class conditional frame probabilities in the specific decision window. The average identification rates are comparable with the rates that are presented in [6], where the best identification rates in [6] are reported as 61% for 5-mixture GMM classifier with 1s decision window. However, we observed significant improvements on the correct classification rates for longer decision window sizes and for increasing number of Gaussian mixtures. Note that the training and testing scenario TTB suffers from the lack of available training data and the identification performances for the TTB results lower than the TTA training and testing scenario. In TTB case available training data is lower than in TTA, since increasing the training data improves the classification performance. Also, the classification performances with TTA and TTB saturate for GMM classifiers with more than 64 and 16 mixtures, respectively.

5.1 Results on Boosting GMM Classifiers

Boosting the GMM classifiers are implemented as described in Section 4.3. The GMM(64) and GMM(8) classifiers are boosted respectively with the two different training and testing scenarios, TTA and TTB, to observe the possible performance gain of boosting GMM classifiers. The correct identification rates for 1s and 3s decision windows upto 5 boosting

iterations are given in Table 5.2. After the second iteration of boosting GMM(64) classifiers, a performance gain, which is better than GMM(128) classifier provides, is achieved. A similar performance gain is also observed with the GMM(8) classifiers under TTB training and testing. Especially, when the challenging automatic music genre classification task is considered with 70% human identification rate over 3s decision windows and 61% identification rate reported in [6], the incremental gain in identification performance is significant.

Table 5.2: Average identification rates for 1s and 3s decision windows with boosting of 64-mixture and 8-mixture GMM classifiers, respectively with TTA and TTB training and testing scenarios.

Average Identification Rates (%)				
Iteration	TTA GMM(64)		TTB GMM(8)	
	1 s	3 s	1 s	3 s
1	69.54	72.64	55.56	57.45
2	69.54	72.64	56.31	58.46
3	70.53	73.64	56.36	58.25
4	70.57	73.40	56.43	58.26
5	70.56	73.75	56.43	58.17

The corresponding confusion matrix after 5 boosting iterations of 64-mixture GMM classifier is given in Table 5.4. When compared with the GMM(64) classifier confusion matrix in Table 5.3, one can observe the increase in correct classification rates for most of the musical genre types, especially classic, disco and rock genres.

In order to verify the functionality of boosting with GMM classifiers, the GMM(8) classifier is boosted over 10 iterations, and both training and testing error rates for 3s decision windows are plotted in Fig. 5.1. Both the training and the testing error rates fall, but training error rate falls sharper as expected. These error rate improvements indicate proper functionality of boosting over GMM classifiers.

	cl	co	di	hi	ja	me	po	re	ro
cl	85.7	0.0	1.5	2.7	7.7	0.0	0.0	0.0	2.4
co	2.3	35.7	20.2	1.3	16.6	0.0	15.7	1.4	6.8
di	0.0	8.6	70.8	0.0	0.0	0.0	20.6	0.0	0.0
hi	1.9	0.0	0.0	88.4	0.4	2.4	0.0	5.6	1.4
ja	12.9	0.7	0.5	0.0	73.3	0.0	0.0	7.8	4.9
me	0.0	3.0	4.4	11.6	0.0	73.1	1.2	0.0	6.6
po	0.9	9.7	13.4	0.0	1.6	0.0	70.6	2.8	1.1
re	0.0	0.0	0.0	4.9	3.3	0.0	0.0	91.5	0.4
ro	0.6	0.0	0.0	6.9	2.6	5.8	0.0	19.4	64.7

Table 5.3: Music genre confusion matrix under TTA training and testing with 64-mixture GMM classifier over 3s decision windows without boosting approach. The music genre types are classical (cl), country (co), disco (di), hiphop (hi), jazz (ja), metal (me), pop (po), reggae (re) and rock (ro).

5.2 Results on Inter-Genre Similarity

The experimental results of this section is generated using TTB training and testing scenario over the music genre classifiers that are employing flat or hierarchical structures with or without the discriminative IGS clustering. The number of final clusters is an important parameter of the hierarchical auto-clustering. Table 5.5 presents the average correct classification rates for varying number of clusters in the hierarchical structure. In the presence of the nine music genre types in our database, the best hierarchical classification rates for both the training and testing databases are achieved with 5-class auto-clustering. The structure of the best hierarchical auto-clustering is given in Fig. 5.2.

The performance comparisons of the flat and hierarchical classifiers with and without the discriminative IGS clustering for varying decision window sizes are given in Table 5.6. Note that as expected the correct classification rates are increasing with the increasing decision window size for all classifiers. The discriminative IGS clustering over the nine music genre types results with some incremental classification gain over the flat classifiers as presented in the IGS column in Table 5.6. The performance of the hierarchical auto-clustering without and with IGS are given in the last two columns with some additional identification improvements.

	cl	co	di	hi	ja	me	po	re	ro
cl	88.7	0.0	0.9	3.4	4.7	0.0	0.0	0.0	2.3
co	2.3	36.3	23.1	0.7	12.8	0.0	18.5	1.3	5.1
di	0.0	6.0	75.6	0.0	0.0	0.2	18.2	0.0	0.0
hi	2.1	0.0	0.0	87.1	0.4	3.7	0.0	5.5	1.3
ja	12.8	0.8	0.5	0.0	72.6	0.0	0.0	9.0	4.4
me	0.0	4.3	3.5	12.7	0.0	74.5	0.6	0.0	4.4
po	0.7	8.7	14.8	0.0	1.3	0.1	70.2	2.2	2.0
re	0.0	0.0	0.0	5.1	3.3	0.0	0.0	91.1	0.6
ro	0.6	0.0	0.0	4.7	2.3	4.7	0.0	20.1	67.7

Table 5.4: Music genre confusion matrix under TTA training and testing with 64-mixture GMM classifier over 3s decision windows with boosting approach. The music genre types are classical (cl), country (co), disco (di), hiphop (hi), jazz (ja), metal (me), pop (po), reggae (re) and rock (ro).

Table 5.5: The average correct classification rates of the M-class hierarchical auto-clustering in the training and testing databases using 8-mixture GMM classifiers with 3s decision windows.

M	9	8	7	6	5	4	3	2
Test	57.47	58.79	59.88	63.06	**66.64**	63.65	61.81	61.17
Train	93.79	93.31	93.05	94.14	**94.28**	94.09	93.32	93.72

The human determined hierarchical classifier, as plotted in Fig. 5.3, is evaluated in Table 5.7. The average classification rates of human determined hierarchical classifier without and with IGS is measured respectively as 68.70% and 75.35% for 20s decision window size. Note that these rates are slightly worse than the hierarchical auto-clustering classifier in Table 5.6. Although the human and machine determined hierarchical structures are quite similar, the classification performance of the auto-clustering has incremental gains for all decision window sizes. The structural closeness and the better performance of the auto-clustering to the human determined hierarchical classifier is a verification of the proposed hierarchical auto-clustering approach.

In Table 5.8, the average correct classification rates for 3s and 20s decision window sizes

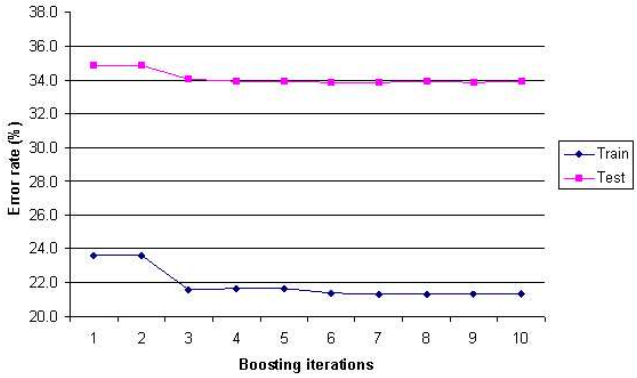

Figure 5.1: Average error rate for train and test data as a function of boosting iterations with GMM(8) classifier.

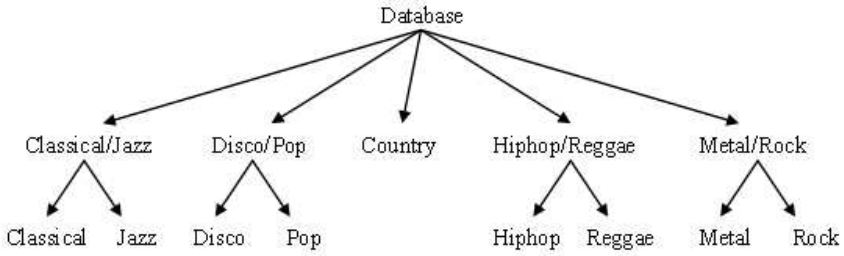

Figure 5.2: The best hierarchical auto-clustering structure.

are given for varying number of GMM mixtures. The correct classification rates increase by increasing number of Gaussian mixtures. However, the classification performance saturates with the 32 mixture Gaussian models. Note that, 13.79% and 14.67% improvements on classification rates are observed with 16 mixture GMM classifiers, respectively for 3s and 20s decision windows. These improvements are significant when compared with the recently presented hierarchical classifier improvements, which are less than 3%, in [18]. Note that, these identification rates are also superior to the performances of the boosting of GMM classifiers that are presented in Table 5.2.

In Table 5.9 and Table 5.10 confusion matrices of hierarchical auto-clustering method are used for music genre classification without and with IGS approach respectively. Diagonal entries shows the correct classification rates. Improvements are higher than the flat

Table 5.6: The average correct classification rates of the flat (F), IGS clustering, hierarchical auto-clustering (HAC), and hierarchical auto-clustering with IGS classifiers using 16-mixture GMM modeling for varying decision window sizes under TTB training and testing scenario.

Decision Window	Correct Classification Rates (%)			
	F	IGS	HAC	HACIGS
10ms	37.96	34.76	39.54	42.87
5ms	55.58	59.72	61.67	65.88
1s	57.91	63.67	65.30	67.83
3s	60.14	69.86	67.98	73.93
20s	63.33	74.18	71.18	78.00

Table 5.7: The average correct classification rates of the human determined hierarchical classifier with and without IGS clustering using 16-mixture GMM modeling for varying decision window sizes.

Decision Window	Correct Classification Rates (%)	
	HC	HCIGS
3s	67.12	69.66
20s	68.70	75.35

distribution case as expected.

In database DB, inter-genre similarity method is tested with flat and hierarchical approaches. For 16 and 32-mixture GMM models, results are given in Table 5.11 and Table 5.12. Increasing window size and increasing number of mixtures improve the classification performance as observed in database DA.

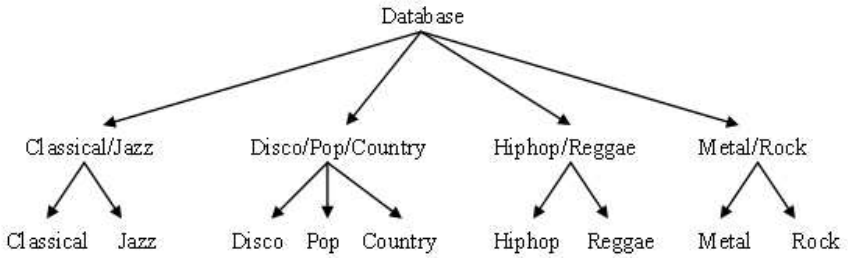

Figure 5.3: The human determined hierarchical clustering structure.

Table 5.8: The average correct classification rates of the flat (F), IGS clustering, hierarchical auto-clustering (HAC), and hierarchical auto-clustering with IGS classifiers using 3s and 20s decision windows for varying number of GMM mixtures.

Dec Win	# of Mix	Correct Classification Rates (%)			
		F	IGS	HAC	HACIGS
3s	8	57.47	62.83	66.64	70.74
	16	60.14	69.86	67.98	73.93
	32	59.26	71.16	69.33	72.92
20s	8	61.67	72.65	68.63	74.94
	16	63.33	74.18	71.18	78.00
	32	61.11	76.20	70.03	80.83

Table 5.9: Music genre confusion matrices of the flat using 16 mixture GMM modeling with 3s decision windows for music genre types country (co), classical (cl), jazz (ja), disco (di), pop (po), rock (ro), metal (me), hiphop (hi) and reggae (re).

Flat Classifier									
	co	cl	ja	di	po	ro	me	hi	re
co	**32.97**	0.051	9.70	20.60	12.06	10.92	0	3.85	9.87
cl	0.53	**79.80**	12.6	0.15	0.91	2.53	0	1.87	1.60
ja	6.35	11.24	**55.78**	0	8.23	12.44	0	1.62	4.33
di	9.72	0.00	0.00	**59.58**	18.95	0.00	0.53	8.47	2.75
po	10.73	0.26	3.62	22.45	**33.75**	6.41	0.53	2.73	19.50
ro	17.9	4.59	7.58	0.43	8.18	**20.57**	9.36	15.93	15.46
me	5.31	0.00	0.00	6.08	0.40	6.74	**64.76**	16.71	0.00
hi	0.45	2.43	0.15	2.19	3.96	2.13	0.85	**69.86**	17.98
re	1.39	0.20	0.67	0.71	0.98	2.40	0.00	11.47	**82.17**

Table 5.10: Music genre confusion matrices of the the hierarchical auto-clustering with IGS classifiers using 16 mixture GMM modeling with 3s decision windows for music genre types country (co), classical (cl), jazz (ja), disco (di), pop (po), rock (ro), metal (me), hiphop (hi) and reggae (re).

HACIGS Classifier									
	co	cl	ja	di	po	ro	me	hi	re
co	**62.67**	0.00	3.09	24.52	1.36	3.54	0.00	1.32	3.49
cl	0.00	**92.54**	4.68	0.00	0.00	0.42	0.00	2.21	0.14
ja	7.35	7.53	**81.37**	0.00	0.00	3.32	0.00	0.00	0.42
di	19.74	0.00	0.00	**70.85**	3.83	0.00	0.46	5.13	0.00
po	17.51	0.00	2.25	30.22	**28.75**	0.00	1.50	0.75	19.02
ro	30.85	1.68	9.13	1.36	0.34	**15.21**	17.29	9.28	14.87
me	1.41	0.00	0.00	3.15	0.00	3.28	**84.09**	8.07	0.00
hi	0.00	2.91	0.00	1.20	0.00	1.94	0.24	**82.49**	11.21
re	0.00	0.00	0.00	0.00	0.00	0.00	0.00	2.38	**97.62**

Table 5.11: Music genre classification over 16-mixture GMM models in DB

DB database is used				
Window Size(ms)	Flat	IGS	HAC	HACIGS
1	37.95	34.76	40.01	42.95
50	55.58	59.72	63.60	68.27
100	57.91	63.67	66.70	69.52
300	60.14	69.86	68.50	75.36

Table 5.12: Music genre classification over 32-mixture GMM models in DB.

DB database is used				
Window Size(ms)	Flat	IGS	HAC	HACIGS
1	38.53	34.79	41.97	44.93
50	56.17	61.69	65.61	69.75
100	57.71	65.35	67.95	72.83
300	59.26	71.16	69.40	74.26

Chapter 6

CONCLUSION

Automatic music genre classification is an important tool for music information retrieval systems. Feature extraction and the classification design are the two significant problem of music genre classification systems. In feature extraction, a set of widely used timbral features are considered.

In this thesis, we investigate two novel classifier structures for discriminative music genre classification. Classification error rates are reduced using the proposed modified EM algorithm for boosting GMM classifiers. It is encouraging that boosting GMM classifiers yields performance gain, which is not attainable with increasing number of Gaussian mixtures. Modified EM algorithm with boosting concept lead us to propose discriminative selection of frames and modeling inside each genre or among the genre pool including mis-classified features from all genres. Increasing number of mixtures or pure boosting methods improve the performances to some extent but at the cost of computational complexity. Boosting with modified EM algorithm reduces the error rate according to flat distribution of music genre classification. It is shown that confusion between genres can be reduced by re-weighting the mis-classified frames and modeling them again. Note that, in modified EM algorithm the weighted responsibilities de-emphasize the easy-to-classify samples, hence hard-to-classify samples are better modeled by the resulting Gaussian mixture density function.

In the second proposed classifier structure the inter-genre similarities are captured and modeled over the mis-classified feature population for the elimination of the inter-genre confusion, and an auto-clustering scheme for hierarchical classification is devised. Experimental results with promising identification improvements, which are superior than the recent literature, are provided. The proposed IGS model provides a decision structure where normalization of the decision probability after removing the probable mis-classified frames helps to improve on classification rate.

After frame based level of similarities, genre based similarities are devised with auto-

clustering techniques by looking at the confusion matrix of classification system. These similarities are the basis of hierarchical clustering in music genre classification. The structure of the music genre hierarchy could be human determined for the flexible browsing or automatically determined for maximizing the classification performance. It is expected to extract similar hierarchy structures for both human and machine determined systems, however the structures may differ due to the different objectives, where in the human determined system ease of browsing and in the machine determined system classification performance are favored. In this thesis, a new auto-clustering algorithm for the hierarchical structure extraction is proposed. The auto-clustering algorithm is iterative and depends on the confusion matrix, where it generates an hierarchy tree by combining the most confused two clusters in each iteration. Since classifiers are concentrating on smaller set of genres that confuse with each other, it is observed that classification rate increases.

Although, the proposed hierarchical auto-clustering with IGS classifier achieves significant identification improvements and results close to the human identification rates for 3s decision windows, we should note that the automatic music genre classification still has a big room for possible improvements. Discriminative feature selection, iterative inter-genre similarity modeling, IGS with soft-weighting and introducing sub-genre definitions to fine represent available databases are planned as the extension of this thesis.

BIBLIOGRAPHY

[1] J.-J. Aucouturier and F. Pachet, "Representing musical genre: A state of the art," *Journal of New Music Research*, vol. 32, no. 1, pp. 83–93, March 2003.

[2] D. Perrot and R. Gjerdigen, "Scanning the dial: An exploration of factors in identification of musical style," in *Proc. Soc. Music Perception Cognition*, 1999, p. 88.

[3] T. Li, M. Ogihara, and Q. Li, "A comparative study on content-based music genre classification," in *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in information retrieval*, 2003, pp. 282–289.

[4] S. Lippens, J.P. Martens, T. De Mulder, and G. Tzanetakis, "A comparison of human and automatic musical genre classification," in *Proc. of the Int. Conf. on Acoustics, Speech and Signal Processing 2004 (ICASSP 2004)*, May 2004, vol. 4, pp. 233–236.

[5] D. Pye, "Content-based methods for managing electronic music," in *Proc. of the Int. Conf. on Acoustics, Speech and Signal Processing 2000 (ICASSP 2000)*, 2000.

[6] G. Tzanetakis and P. Cook, "Musical genre classification of audio signals," *Speech and Audio Processing, IEEE Transactions on*, vol. 10, no. 5, pp. 293–302, July 2002.

[7] C. Xu, N.C. Maddage, X. Shao, F. Cao, and Q. Tian, "Musical genre classification using support vector machines," in *Proc. of the Int. Conf. on Acoustics, Speech and Signal Processing 2003 (ICASSP 2003)*, April 2003, vol. V.

[8] U. Bağcı and E. Erzin, "Boosting classifiers for music genre classification," in *20th International Symposium on Computer and Information Sciences (ISCIS 2005) and also to appear in Lecture Notes in Computer Science (LNCS) by Springer-Verlag*, Istanbul, October 2005.

[9] Xi. Shao, Changsheng. Xu, and S. Mohan Kankanhalli, "Unsupervised classification of music genre using hidden markov model," in *Proc. of the IEEE International Conf. of Multimedia Expo*, 2004.

[10] G. Tzanetakis, "Manipulation, analysis and retrieval systems for audio signals," *PhD Dissertation*, 2002.

[11] W. J. Dowling and D. L. Harwood, *Music Cognition*, Academic Press, Inc, 1986.

[12] J. Foote, "Content-based retrieval of music and audio," *Multimedia Storage and Archiving Systems II, Proceedings of SPIE*, pp. 138–147, 1997.

[13] T. Lambrou, P. Kudumakis, M. Sandler, R. Speller, and A. Linney, "Classification of audio signals using statistical features on time and wavelet transform domains," in *Proc. of the Int. Conf. on Acoustics, Speech and Signal Processing 1998 (ICASSP 1998)*, 1998.

[14] H. Soltau, "Erkennung von musikstilen," *Master of Science Thesis*, 1997.

[15] H. Soltau, "Recognition of music types," in *Proc. of the Int. Conf. on Acoustics, Speech and Signal Processing 1998 (ICASSP 1998)*, 1998.

[16] A. Meng, P. Ahrendt, and J. Larsen, "Improving music genre classification by short-time feature integration," in *Proc. of the Int. Conf. on Acoustics, Speech and Signal Processing 2005 (ICASSP 2005)*, Philadelphia, March 2005, vol. V, pp. 497–500.

[17] Richard O. Duda, Peter E. Hart, and David G. Stork, *Pattern Classification(2nd Edition)*, Wiley-Interscience, 2000.

[18] T. Li and M. Ogihara, "Music genre classification with taxonomy," in *Proc. of the Int. Conf. on Acoustics, Speech and Signal Processing 2005 (ICASSP 2005)*, Philadelphia, March 2005, vol. V, pp. 197–200.

[19] T. Heittola, "Automatic classification of music signals," *Master of Science Thesis*, 2003.

[20] J.J. Burred and A. Lerch, "A hierarchical approach to automatic musical genre classification," in *International Conference on Digital Audio Effects (DAFx03)*, London, UK, September 2003, pp. 308–311.

[21] D.A. Reynolds and R.C. Rose, "Robust text-independent speaker identification using gaussian mixture speaker models," *Speech and Audio Processing, IEEE Transactions on*, vol. 3, no. 1, pp. 72–83, 1995.

[22] A. Acero X. Huang and H. W. Hon, *Spoken Language Processing: a guide theory, algorithm and system development*, Prentice Hall PTR, 2001.

[23] F.T. Quatieri, *Discrete-Time Speech Signal Processing Principles and Practice*, Prentice Hall, 2001.

[24] M. Slaney, "Construction and evaluation of a robust multifeature speech/music discriminator," in *Proc. of the Int. Conf. on Acoustics, Speech and Signal Processing 1997 (ICASSP 1997)*, 1997, pp. 1331–1334.

[25] M. Slaney, "Auditory toolbox, version 2. technical report," in *Interval Research Corporation*, 1998, number 1998-010.

[26] E. Alpaydin, *Introduction to Machine Learning*, The MIT Press, 2004.

[27] L. Schwardt, "Gaussian mixture models," in *TW414/PR813 Lecture Notes*, University of Stellenbosch, 2003.

[28] R. E. Schapire, "A brief introduction to boosting," in *In Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence*, 1999.

[29] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," *Journal of Computer and System Sciences*, vol. 55, no. 1, pp. 119–139, 1997.

[30] R. A. Redner and H. F. Walker, "Mixture densities, maximum likelihood and the EM algorithm," *SIAM Rev.*, vol. 26, no. 2, pp. 195–239, 1984.