Abstract

The Internet has revolutionized the music industry. The onus has shifted from physical media to online products and services. As a consequence, a great quantity of music is now available online. However, online music is still classified into genres manually, by experts. This is both time-consuming and costly. Thus, there is a need for music classification and discovery technologies that allow users to explore the various options available online. Digital music distribution giants such as iTunes or Amazon provide quick access to a massive amount of music at low prices, so they are less strictly filtered. Also, with the decline of physical records, they shifted focus from albums to single tracks, making it even harder for users to find songs they like. The aim of this project is to improve upon the pre-existing genre classification algorithms by using audio similarities. Each genre has certain commonalities, of style, content and mood, unique to that particular type of music. Our goal is to extract those common features, identify them, modify pre-existing methods for identification of the aforementioned to increase efficiency and scope of implementation, and then use them to classify musical pieces/songs according to genre. We also hope to implement the `search-by-example' system, which recommends music to users and generates playlists.

Contents

List of Figures

List of Tables

List of Abbreviations

MFCC: **M**el **F**requency **C**epstral **C**oefficient

DCT: **D**iscrete **C**osine **T**ransform

RMS: **R**oot **M**ean **S**quare

FFT: **F**ast **F**ourier **T**ransform

GUI: **G**raphical **U**ser **I**nterface

SVM: **S**upport **V**ector **M**achine

MATLAB: **MAT**rix **LAB**oratory

Avg.: Average

Chapter 1

Introduction

Music can be divided into categories, based on the style, rhythm, and even culture. The different styles are what we call “genres”. Musical genres are, however, subjective, and one song can consist of several different styles with different weightages. A lot of countries and organizations define the same genres differently. There is no official definition of music genres till now.

Currently, genres are tagged by musical experts - musicians, professors and artists - and thus the process of classification is tainted by their individual cultural heritage. Also, the number of musical pieces that need tagging runs into millions at the very least. Thus, an objective and automated genre classification system is desperately needed. There are hundreds of genres in music. Different genres may seem to be the same at first listen, so classification is a very complex task.

In this project, only three genres are used for simplicity and practicality. They were chosen because they differ from each other considerably. Also, these genres have been around for a long period of time, and various diverse artists have contributed to them. A lot of other genres did not make the cut because they are defined much more by the music’s purpose and usage than the actual musical content. They traverse multiple other genres, and the songs are not only amongst an album, but within any given track (e.g. - jazz, hard rock). The inclusion of these categories was decided to be too ambitious. Beyond this, the decision was essentially made from whatever high quality music was legally available.

The genres that were selected were:

- **Western Classical**

The term Western Classical is used loosely here, that is, any ‘older’ sounding music of an orchestral, instrumental or operatic persuasion. It includes the Baroque, Classical, Romantic and Modern eras.

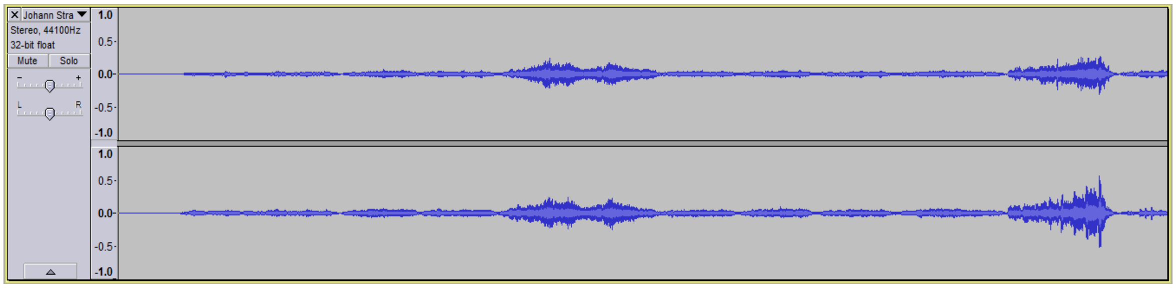

Fig 1.1: A one minute excerpt of Blue Danube Waltz, By Johann Strauss II

- **Metal**

This category is a collection of the earliest forms of metal – Thrash Metal (Characterized by shredding, aggressive playing, fast tempos, anti-establishment lyrics and the use of two bass drums) and Heavy Metal (A sub-genre of rock characterized by loud, distorted guitars and dark lyrical matter).

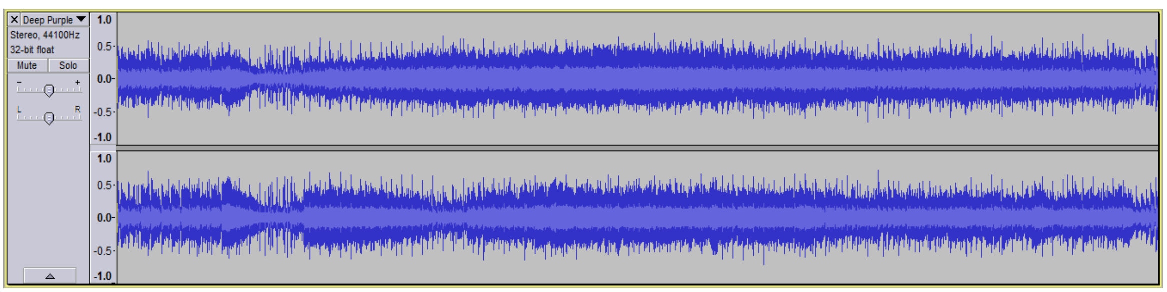

Fig 1.2: A one minute excerpt of N.I.B., by Black Sabbath

- **Blues**

This category consists of early acoustic blues, early Chicago blues and modern blues rock.

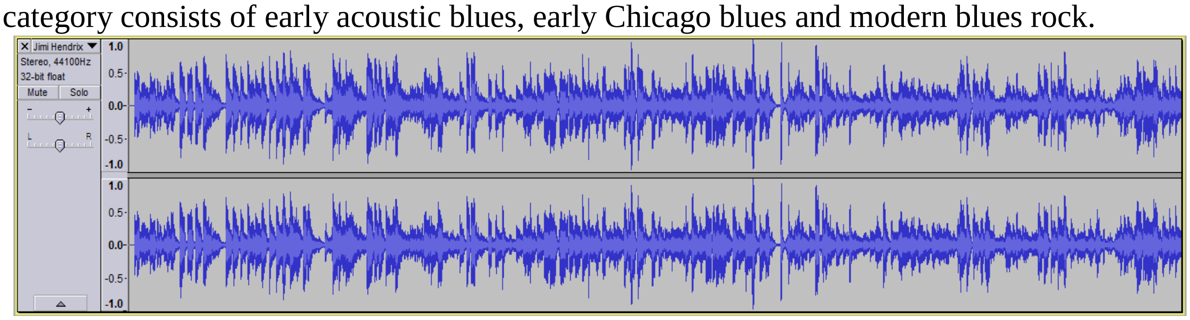

Fig 1.3: A one minute excerpt of Hear My Train Comin’, by Jimi Hendrix

Chapter 2

Literature Review

In this chapter, we'll talk briefly about the paper that we read during our research. These papers have, in turn, helped us form a clear picture of the path we needed to take for our project.

2.1 Musical Genre Classification of Audio Signals, by George Tzanetakis and Perry Cook

This paper was one of the first to talk about automatic genre classification. Using a 30 dimensional feature vector (9 FFT +10 MFCC +6 rhythmic content features+5 pitch content features), they achieved an accuracy of 61% in non real time and 44% in real time. It was pointed out that if real time performance is desired, then only timbral texture features can be used.

2.2 Mel Frequency Cepstral Coefficients for Music Modeling, by Beth Logan

This paper investigates the applicability of MFCCs to music modeling by focusing on two main assumptions of the process – the use of mel frequency scale to model the spectra, and the use of Discrete Cosine Tranform to decorrelate the mel-spectral vectors.

It was found the use of mel scale for music modeling is at least not harmful, and that the use of DCT to decorrelate vectors is appropriate for both speech and music spectra.

2.3 Music Genre Classification with the Million Song Dataset, by Dawen Liang, Haijie Gu and Brendan O'Connor

The authors proposed a novel cross-modal retrieval framework that blended lyrical and audio features for automated classification of the Million Song Dataset, consisting of 10,00,000 songs by 44,745 unique artists. This paper was unique in its use of bag-of-words lyric features for genre classification, as lyrical content often defines genres.

They best accuracy of 38.1% was achieved by a combination of Baum-Welch HMM, Spectral HMM, loudness, tempo and lyrics. They hypothesized that the pop and rock genres are typified less by musical features and more by cultural style and historical periods.

2.4 A MATLAB Toolbox to compute music similarity from audio, by Elias Pampalk

This paper presented a MATLAB toolbox implementing music similarity measures for audio. It implemented frame clustering, cluster model similiarity, spectrum histograms, periodicity histograms, as well the Islands of Music visualization.

2.5 Automatic Classification of Music Signals, by Toni Heittola

This thesis studied content-based classification of music signals according to genres, and the instruments used, and also proposed a method for detection of drums based on the periodicity of the amplitude envelopes of the signal at subbands. A listening test was conducted to study the human ability to recognize genres, with humans being able to recognize the correct genre in 75% of the cases (given five second samples).

Detection accuracy of 60% was obtained by the developed automatic recognition system, along with an accuracy of 81% obtained by the proposed drum detection method.

2.6 Indexing Content-Based Music Similarity Models for Fast Retrieval in massive databases, by Dominik Schnitzer

This thesis developed a large-scale music recommendation system, by using non-vectorial music similarity features with their non-metric divergences in centroid-computing algorithms. It also put forward a new method to speed up music recommendation queries, by using a filter-and-refine systems layout. The developed prototype, 'Wolperdinger', operates on a collection of 2.3 million songs and is able to answer recommendation queries in a fraction of a second. It is the largest content-based music recommendation system published to date.

2.7 Unsupervised Audio Feature Extraction for Music Similarity Estimation, by Jan Schluter

This thesis covers the development of a music similarity estimation system based on unsupervisedly learnt audio features. A recently proposed generative model, the mean covariance Restricted Boltzmann Machine (mcRBM), was trained on music spectrogram excerpts and employed for feature extraction. It clearly outperformed the classic MFCC-based method and surpasses simple unsupervised feature extraction on three public datasets.

2.8 Clustering and Classification Techniques

Clustering Algorithms, by Jure Leskovec and Anand Rajaraman

The authors cover multiple clustering algorithms over the course of this presentation. Starting with a problem statement, it then goes on to state a few applications of clustering, like SkyCat, a catalog of 2 billion objects that represent objects by their radiation into 7 frequency bands. The object is to classify similar objects like galaxies and stars. Then, it moves on to the different methods of clustering, hierarchical clustering and point assignment. It covers hierarchical clustering, k-means clustering, CURE (Clustering Using REpresentatives) algorithm and Bradley-Fayyad-Reina (BFR) algorithm. It also implements SkyCat using K-means.

Comparisons Between Data Clustering Algorithms, by Osama Abu Abbas

This paper takes four different clustering algorithms for investigation, study and comparison – K means algorithm, hierarchical clustering algorithm, Self-Organization Map (SOM) and Expectation Maximization (EM) clustering algorithm.

The conclusions were that k-means and EM were excellent at clustering bigger datasets, while SOM and hierarchical clustering were better at clustering smaller datasets. Also, all the algorithms showed ambiguous results when noisy data was clustered.

2.8.1 Support Vector Machine (SVM)

Support Vector Machine – Concept and MATLAB Build, by Kan Xie

In this paper, the author talks about, and implements, a simple Support Vector machine. The implementation is done in MATLAB. SVM is a machine learning method that is used for pattern recognition and classification. The algorithm was invented by Vladimir Vapnik. It creates a hyperplane in between sets of data in order to indicate which class it belongs to. For good results, the hyperplane should be at a large distance from the nearest training data points of any class.

2.8.2 K-Means

Selection of K in K-means clustering, by D.T. Pham, S.S. Dimov and C.D. Nguyen

This paper proposed a new method to select the number of clusters for k-means algorithm. The method is calls for a number of test runs for multiple values of k automatically. The optimum value of k, or the number of clusters, is then found from the k values for the test runs, and suggested.

The K-Means Algorithm, from Introduction to Data Mining by Tan, Steinbach, Kumar and Ghosh

The authors talk about how the k-means clustering algorithm works. They point out the limitations of the technique – it doesn't work for non-globular clusters and clusters of differing sizes and densities. It also covers the area of k-means research, and talks about its multiple modifications like fuzzy c-means and harmonic k-means. It also covers the steps of the algorithm in some detail.

2.9 Summary

Music genre classification is a fundamental component of music information retrieval systems and has been gaining importance and enjoying a growing amount of attention with the emergence of music on the Internet. . Each genre has certain commonalities, of style, content and mood, unique to that particular type of music. Our goal is to extract those common features, identify them, modify pre-existing methods for identification of the aforementioned to increase efficiency and scope of implementation, and then use them to classify musical songs according to genre. Since the rapidly increasing size of database, digital music becomes much more popular. An automatic and faster system is thus demanding for enhancing the searching efficiency and quality and music genre classification.

Chapter 3

Project Design

3.1 Motivation

The current method used for genre classification is hopelessly outdated. Musical experts-scientists, researchers and journalists- are currently used to add metadata to the already massive amount of music available online. Not only does this method take a lot of time, but the metadata often added is inaccurate due to the subjectivity of the genres. This led us to try and create an objective Music Classification System. This would not only make it faster but it would also allow multiple/large number of genres to be classified simultaneously, thus saving both time and labor. The base reason for this problem to be chosen is the appreciation that everybody has for music. Any development in the field of music is noticed and appreciated by all.

3.2 Problem Formulation

The research and study behind this topic can be subdivided into three phases:

- First: Collecting songs for the database for both training and the testing phase for different genres. We collected 15 songs, 5 each for 3 genres- Metal, Blues and Western Classical. Out of these 5 songs, 2 are of training and 3 are for testing. The songs were taken as 1 min clips in .wav format.
- Second: Audio features identification and extraction, which involves identifying and extracting the essential features from a song and then applying signal processing algorithms and techniques in order to analyze these different features.
- Third: K-Means algorithm. This is last and the most important phase where the unknown song is classified according to its features.

3.3 Objective

The aim of this project is to improve upon the pre-existing genre classification algorithms by using audio similarities by extracting the common features, identifying them and modifying pre-existing methods for identification of the aforementioned and then use them to classify musical pieces/songs.

3.4 Methodology Adopted

We started by collecting songs of different genres for our databases we are considering. The next phase was to extract the features. This was done by MIR (Music Information Retrieval), which includes methods like MFCC (Mel Frequency Cepstral Coefficient). The next step was to use these features in conjunction with machine learning techniques, to automatically classify pieces according to the chosen classifying factor. While doing classification, the algorithm runs in the background for finding an estimate of the similarity between two songs and defines a similarity function. To find the music pieces most similar to a given song, in the simplest case a retrieval algorithm linearly scans and compares all music pieces to the query according to the similarity function. The result of classification still remains subjective and influenced by cultural background. Therefore, we will extract meaningful features in order to classify music genre with a more standardized manner. When features of music are extracted, we have a high dimensional feature space to be classified using algorithms with unsupervised or supervised approaches. Then is the testing phase in which we find the accuracy of the songs classified into their genres.

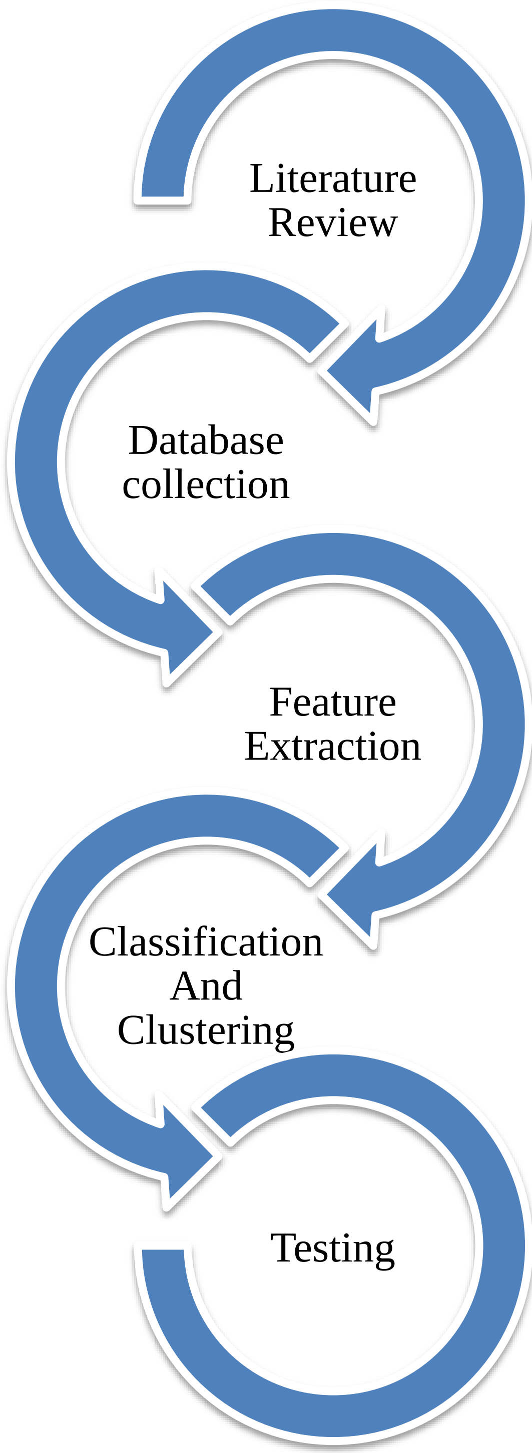

Fig 3.1: Flowchart describing the methodology of our project-

"Evaluation Of Audio Similarities For Genre Based Classification and Recommendation of Music"

Chapter 4
Database Collection

The database consists of three genres – Blues, Metal & Western Classical. These genres consist of 10 samples each – 1 minute long Wave Sound (.wav) files. Each sample has a bit depth of 16 bits, a bit rate of 1411 kbps and a sampling rate of 44.1 KHz. This amounts to a total of 30 samples for the training set. For testing, 15 samples are taken in total, i.e., 5 samples per genre. A list of the songs used, along with a brief explanation of the characteristics of genres, is as follows.

4.1 Blues

Blues, being about tradition and self-expression, still retains the essential features of its infancy. Most blues is marked by simple progressions, with structures that encourage both lyrical and musical improvisation. The blues grew out of the old work songs and spirituals of the African people. These songs have been a long oral tradition. Hybrid forms defined every region, with the earliest recorded blues from the 1900s was dominated by simple sounding acoustic guitars and rustic pianos. Blues fragmented in the 1940s and 50s, when most bluesmen followed Muddy Waters and started playing the blues on electric instruments. From then on, blues has been defined by both the continuing acoustic tradition and the experimental and jazzier electric blues. Blues is defined by the blues notes – ‘flatter’ notes that are thrown in along with the notes of the major scale for emotive purposes. These notes make the blues the powerful music that it is.

Table 4.1: The blues songs used in the database – Training set

Name of the song	Artist
The Thrill is Gone	B.B. King
Ain’t No Sunshine	Bill Withers
Sittin’ On Top Of The World	Howlin’ Wolf
Poor Johnny	Robert Cray
Mad About the Boy	Dinah Washington
How Deep Is the Ocean	Etta James
Cry Me a River	Natalie Cole
Smile	Nat King Cole
The Man I Love	Sarah Vaughan
If I Could	Ray Charles

Table 4.2: The blues songs used in the database – Testing set

Name of the song	Artist
Summertime	Ella Fitzgerald
Unchain My Heart	Joe Cocker
I Put A Spell On You	Nina Simone
Hear My Train Comin’	Jimi Hendrix
Besame Mucho	Nicki Parrott

4.2 Metal

Any loud, distorted guitar driven music is metal music. This category includes the earliest forms of metal – Thrash Metal (Characterized by shredding, aggressive playing, fast tempos, anti-establishment lyrics and the use of two bass drums) and Heavy Metal (A sub-genre of rock characterized by loud, distorted guitars and dark lyrical matter).

Table 4.3: The metal songs used in the database – Training set

Name of the song	**Artist**
N.I.B.	Black Sabbath
Paranoid	Black Sabbath
Number Of The Beast	Iron Maiden
When The Levee Breaks	Led Zeppelin
Sad But True	Metallica
Faries Wear Boots	Black Sabbath
Hard Road	Black Sabbath
Iron Man	Black Sabbath
5 Minutes Alone	Pantera
By Demons Be Driven	Pantera

Table 4.4: The metal songs used in the database – Testing set

Name of the song	Artist
Headless Cross	Black Sabbath
Jerusalem	Black Sabbath
Johnny Blade	Black Sabbath
Juniors Eyes	Black Sabbath
Kill in the Spirit World	Black Sabbath

4.3 Western Classical

Western Classical is an umbrella term used for ‘older’ sounding music from Europe. It is divided into periods, the most notable of which are the Baroque, the Classical and the Modern era. It relies on polyphony as a tool for musical exploration, with emphasis being removed from the rhythm. Often, pieces traverse through different musical landscapes till the central theme or idea is revealed.

Table 4.5: The western classical songs used in the dataset – Training set

Name of the song	Artist
Pie Jesu	Andrew Lloyd Webber
Piano Concerto No. 2	Frederic Chopin
Blue Danube Waltz	Johann Strauss II
Moonlight Sonata, Mvt. 1	Ludwig Van Beethoven
Queen Of The Night Aria	Wolfgang Amadeus Mozart
Clarinet Quintet in A Major	Mozart
Duo No.1 in G Major	Mozart
Flute Quartet No.2 in G Major	Mozart
Piano Sonata No.8 in A Minor	Mozart
Oboe Quartet in F Major	Mozart

Table 4.6: The western classical songs used in the database – Testing set

Name of the song	Artist
Piano Sonata No.16 in C Major	Mozart
Quintet For Piano & Wind in E	Mozart
Serenade No.12 in C Minor	Mozart
Sonata For Two Pianos in D Major	Mozart
String Quartet No.17 in B Flat	Mozart

Chapter 5

Feature Extraction

5.1 Mel Frequency Cepstral Coefficient (MFCC)

Mel Frequency Cepstral Coefficents (MFCCs) are a standard pre-processing technique in speech processing. It was originally developed for automatic speech recognition but have been proven to be useful for music information retrieval. It is one of the simplest ways to represent the timbre of the piece.

Fig 5.1: The flow chart demonstrating how to calculate MFCCs

1. **Pre-Emphasis**:

Pre-emphasis: boosting the energy in the high frequencies.

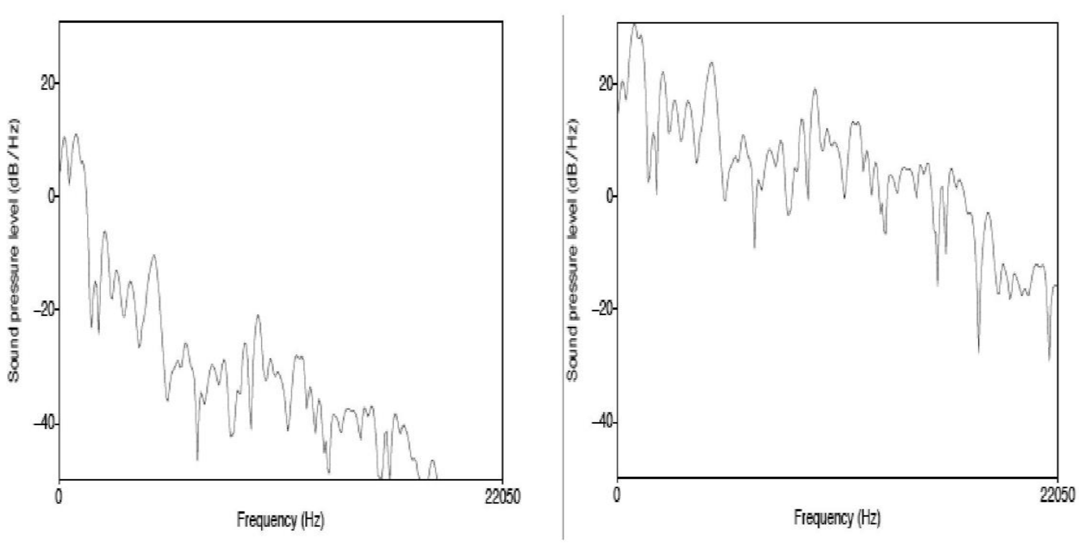

Fig 5.2: Before and after pre-emphasis

2. Windowing:

Speech is not a stationary signal; we want information about a small enough region that the spectral information is a useful hint.

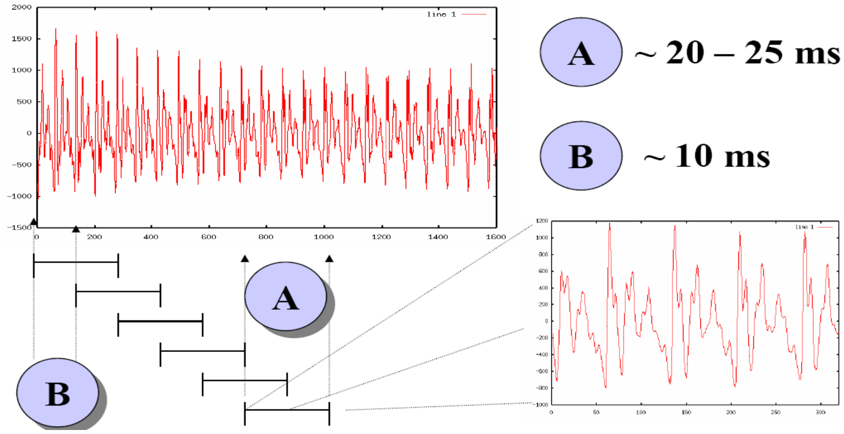

Fig 5.3: Hamming Windowing

3. Discrete Fourier Transform:

Input: Windowed signal x[n]…x[m].

Output: For each of N discrete frequency bands. A complex number X[k] representing magnitude and phase of that frequency component in the original signal.

4. Mel Scale:

The Mel scale relates perceived frequency, or pitch, of a pure tone to its actual measured frequency. Humans are much better at discerning small changes in pitch at low frequencies than they are at high frequencies. Incorporating this scale makes our features match more closely what humans hear.

The formula for converting from frequency to Mel scale is:

$$M(f) = 1125 \ln\left(1 + \frac{f}{700}\right) \tag{1}$$

To go from Mels back to frequency:

$$M^{-1}(m) = 700\left(\exp\left(\frac{m}{1125}\right) - 1\right) \tag{2}$$

$$X[k] = \sum_{n=0}^{N-1} x[n]e^{-j2\frac{\pi}{N}kn} \tag{3}$$

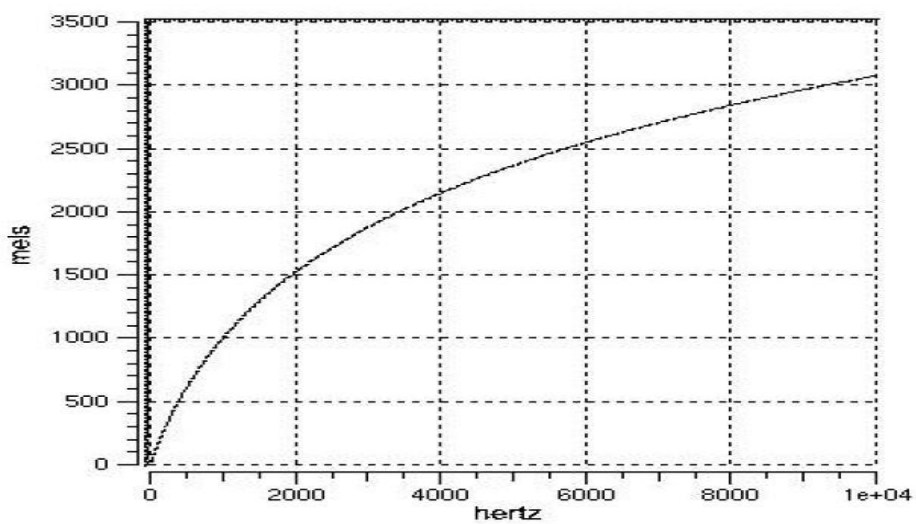

Fig 5.4: Mel scale

5. Mel Filter Bank Processing :

Apply the bank of filters according Mel scale to the spectrum. The triangle bandpass filters in Figure 5 are used for mel-scaling mask smoothing, and are designed based on the sensitivity of hearing of human.Since the perception of hearing for low frequency is much more sensitive than that for high frequency, the smoothing triangle filter smooth larger range of high frequency, while the bandwidth of the smoothing filter for low frequency is narrower. The main purpose of triangle bandpass filters is to emphasize the Formants, i.e. the maximal power in local spectrum, and eliminate the influence of harmonics.

Therefore, MFCCs are independent of the pitch and tone of the audio signal, and thus can be an excellent feature set for speech recognition and audio processing.

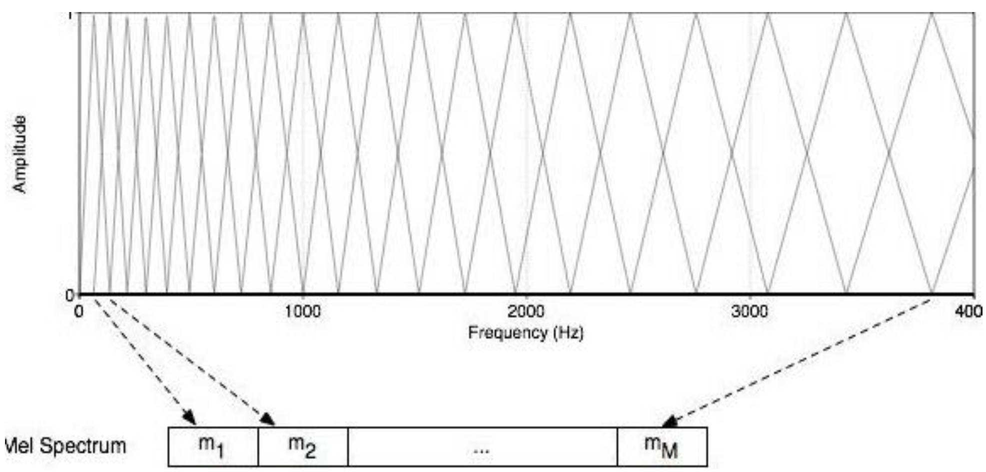

Fig 5.5: Mel Filter Bank Processing (Tribank Filters)

6. Log Energy Computation:

Logarithm compresses dynamic range of values. Log energy of the signal frame and12 coefficients of cepstrum, that is, 13-dimension feature set is the basic MFCCs for an audio signal frame.

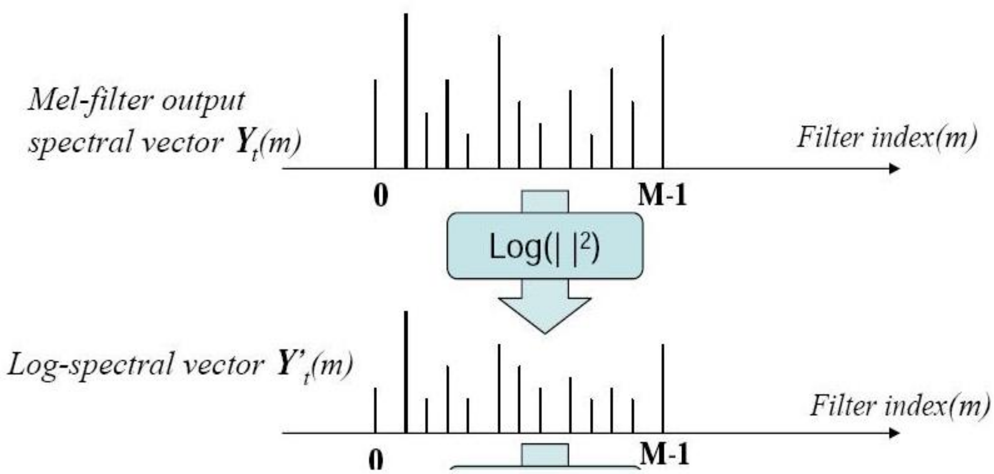

Fig 5.6: Log Energy Computation

7. Applying Discrete Cosine Transform

DCT is applied to the resulting spectrum, as the larger spectral shape is more important than the noise, hidden as details. When DCT is applied and the higher coefficients are discarded, we are left with the smooth spectral shape without much noise.

8. Derivatives:

In order to obtain temporal Information. Also known as differential and acceleration coefficients. The MFCC feature vector describes only the power spectral envelope of a single frame, but it seems like speech would also have information in the dynamics i.e. what are the trajectories of the MFCC coefficients over time. It turns out that calculating the MFCC trajectories and appending them to the original feature vector increases ASR performance by quite a bit (if we have 12 MFCC coefficients, we would also get 12 delta coefficients, which would combine to give a feature vector of length 24).

To calculate the delta coefficients, the following formula is used:

$$d_t = \frac{\sum_{n=1}^{N} n(c_{t+n} - c_{t-n})}{2\sum_{n=1}^{N} n^2} \tag{4}$$

where, d_t is a delta coefficient, from frame t.

t is computed in terms of the static coefficients c_{t+N} to c_{t-N}

N is typically 2.

There are a few more things commonly done, sometimes the frame energy is appended to each feature vector. Delta and Delta-Delta features are usually also appended. Liftering is also commonly applied to the final features.

5.2 Root Mean Square

RMS or Root Mean Square is a measure of amplitude of sound wave in one analysis window. This is defined as:

$$RMS = \sqrt{\frac{x_1^2+x_2^2+\cdots+x_n^2}{n}} \qquad (5)$$

where, n is the number of samples within an analysis window and x is the value of the sample.

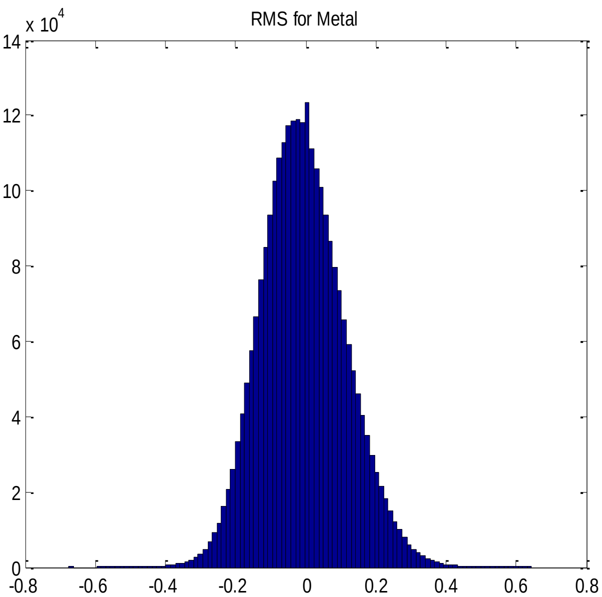

Fig 5.7(a): Metal

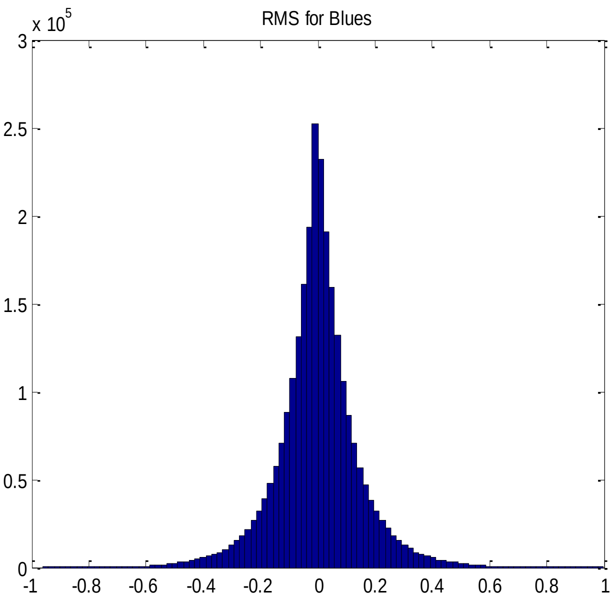

Fig 5.7(b): Blues

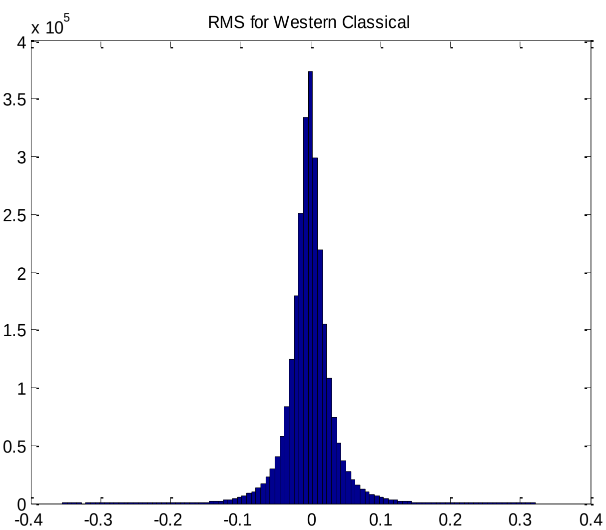

Fig 5.7(c): Western Classical

Fig 5.7: Histograms of audio file, divided into 100 equally spaced bins

5.3 Periodogram

The **periodogram** is an assessment of the spectral density of a signal. It is defined as:

$$I_N(e^{j\omega}) = \frac{1}{N}\left|\sum_{n=0}^{N-1} x[n]e^{-jn\omega}\right|^2 \tag{6}$$

$$I_N(e^{j\omega}) = \frac{1}{N}\sum_{n=0}^{N-1} x[n]e^{-jn\omega}\sum_{r=0}^{N-1} x[r]e^{jr\omega} \tag{7}$$

And distinctly evaluations at $\omega = \omega_k = \frac{2\pi}{N}k$ are efficiently computable via the FFT.

The periodogram is usually calculated from a fixed-length digital sequence using Fast Fourier transform (FFT), as the raw periodogram itself is not a good spectral feature, due to the fact that increasing the number of samples does not decrease the variance of a given frequency.

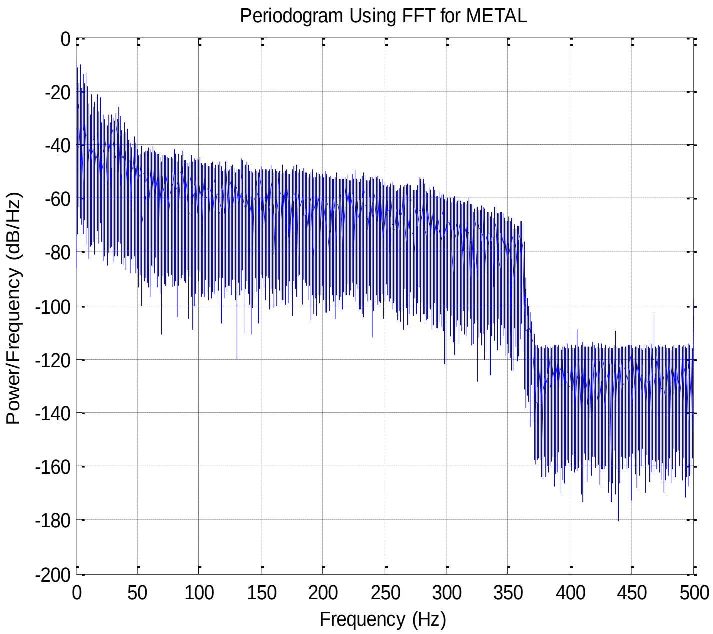

Fig 5.8 (a)

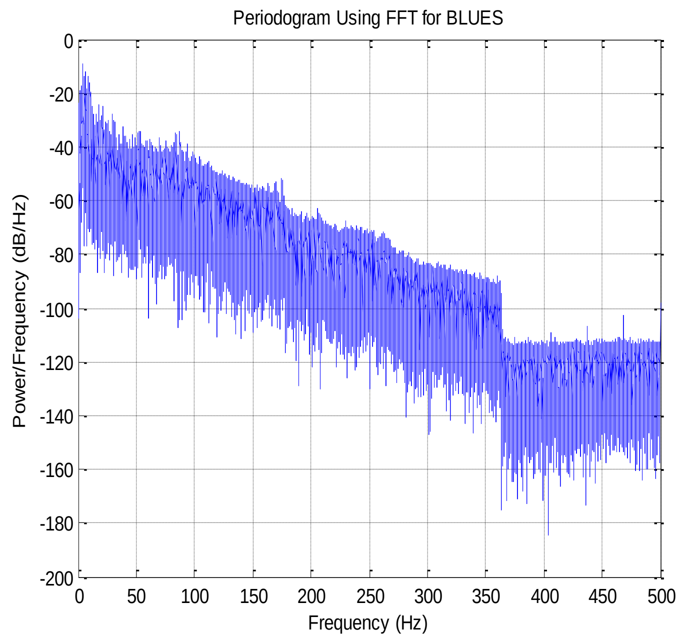

Fig 5.8 (b)

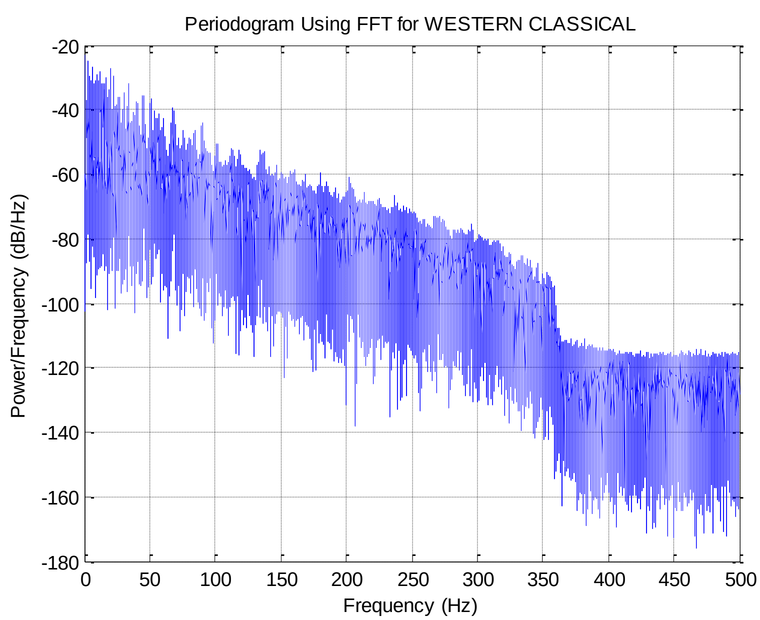

Fig 5.8 (c)

Fig 5.8: Periodogram using FFT for (a) Metal (b) Blues (c) Western Classical

Chapter 6

Clustering and Classification

6.1 K-Means

When features span a high dimensional space, each song is seen as a feature set, i.e. a position in the feature space as figure 6.1(a). The operations of K-means are shown in figure 6.1 (take the 2-D feature space as example).

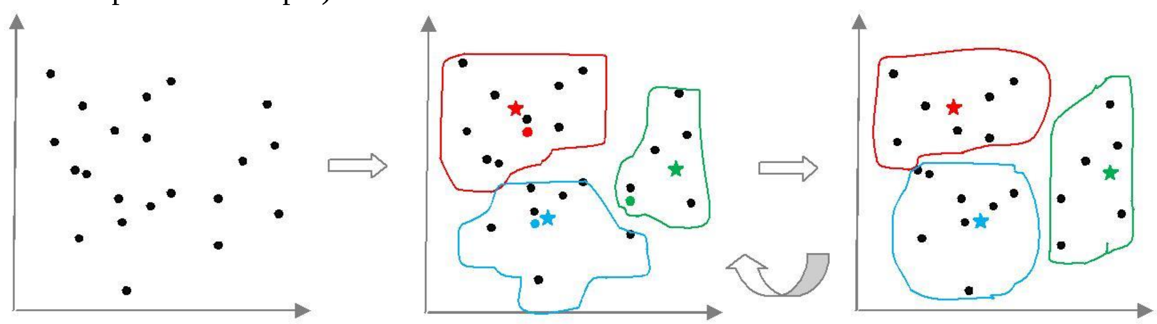

Fig 6.1(a) **Fig 6.1(b)** **Fig 6.1(c)**

Fig 6.1: K-Means Clustering

Initially, the data is scattered in the feature space. K-means take k points (in this example, k=3) as the initialized centroid, and the k centroid are the red, blue and green dots in 6.1(b). Then, each point in feature space is classified into one of the 3 clusters by minimizing the root mean square distance between the centroid and the point. The result of the first iteration is as the red, blue and green boundary in 6.1(b), and this is the new clustering result. Next, we calculate each centroids of the new clusters, repeat the operations of 6.1(b) until the centroids of clusters do not change anymore. The resulting clusters describe the similarity among songs, that is, the songs belong to the same cluster may sound more similar. The drawback of k-means is that "k" must be decided in advance, and how to choose a proper "k" is still an intractable issue.

6.2 Mechanism of K-Means Clustering process

- **Step 1:**

  Decide the number of clusters, symbolized by 'k'.

- **Step 2:**

  Start with any initial partition that classifies the data into 'k' clusters. Assignation of training set can be done randomly or systematically, as given below:

  a. The first 'k' training samples are taken as single-element clusters.

  b. The remaining training samples are assigned to the cluster with the nearest centroid.

  c. The centroid of the growing cluster is recomputed with every assignation.

- **Step 3:**

  Each sample is taken in order and its distance from the centroid of each of the clusters is calculated. If a sample is not currently in the cluster with the closest centroid, then it is switched to that cluster. Then, the centroids of the cluster gaining the sample and the cluster losing it are recomputed.

- **Step 4:**

  Step 3 is repeated till convergence is achieved. Convergence is the state in which passing through the training samples causes no new assignments.

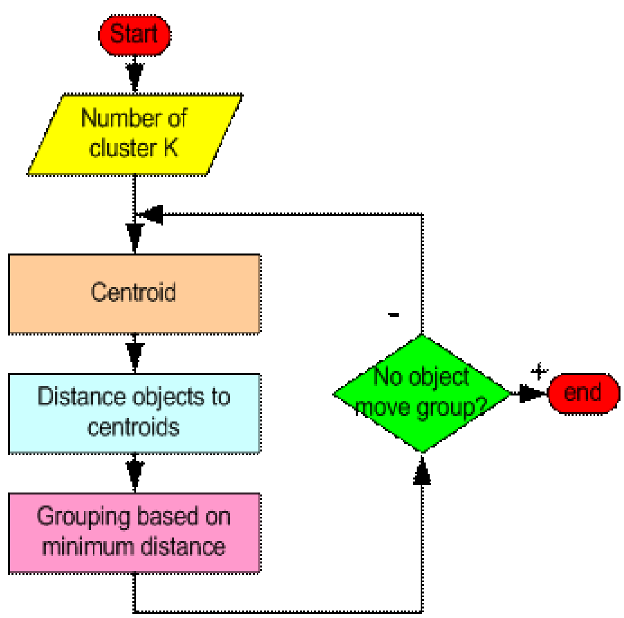

Fig 6.2: Flowchart explaining how K-means work

6.3 Advantages and disadvantages of k-means clustering

Advantages

- When the number of clusters is small and the number of variables involved is large, the K Means outperforms other hierarchical clustering models.
- In the case of globular clusters, K-Means may produce tighter clusters than other hierarchical clustering models.

Disadvantages

- It is difficult to compare the quality of the clusters produced - for different initial partitions or values of K that affect the outcome.
- For non-globular clusters, fixing the number of clusters to a small value affects the outcome negatively.
- Different final clusters can be achieved by taking different initial partitions, or by taking different values of K.

Chapter 7

Conclusions and Results

Genres, while subjective, do have differing audio qualities. Thus, our model does operate, but with a very small database. It is a long way off from being an easily-used system for genre classification. The results are encouraging for the given database, with an accuracy of 60 to 70%

7.1 MFCC

The parameters considered for feature extraction and classification of a certain audio file.

- **Analysis frame duration (Tw) = 60 ms**
  This means dividing the signal into 60 ms frames.

- **Analysis frame shift (Ts) = 55 ms**
  Frame step size of 55 ms allows an overlap of 5 ms between consecutive frames.

- **Pre-emphasis coefficient (alpha) = 0.98**
  This coefficient is used for the boosting the energy of signal at high frequency.

- **Frequency Range = 20 to 20000 Hz**
  This is chosen according to the length of the audio file and the viable frequencies that carry the most information.

- **Number of filter bank channels (M) = 24**

- **Number of cepstral coefficients (C) = 15**

- **Cepstral sine lifter parameter (L) =22**

The results with these parameters were efficient and accurate.

7.1.1 Triangular Filter Bank

The triangular filter bank is applied to the signal so as to convert it to a logarithmic function, which can be divided into mel-spaced bins. It is a technique for obtaining the non-linear resolution and they are spread over the entire frequency range from zero up to the Nyquist frequency.

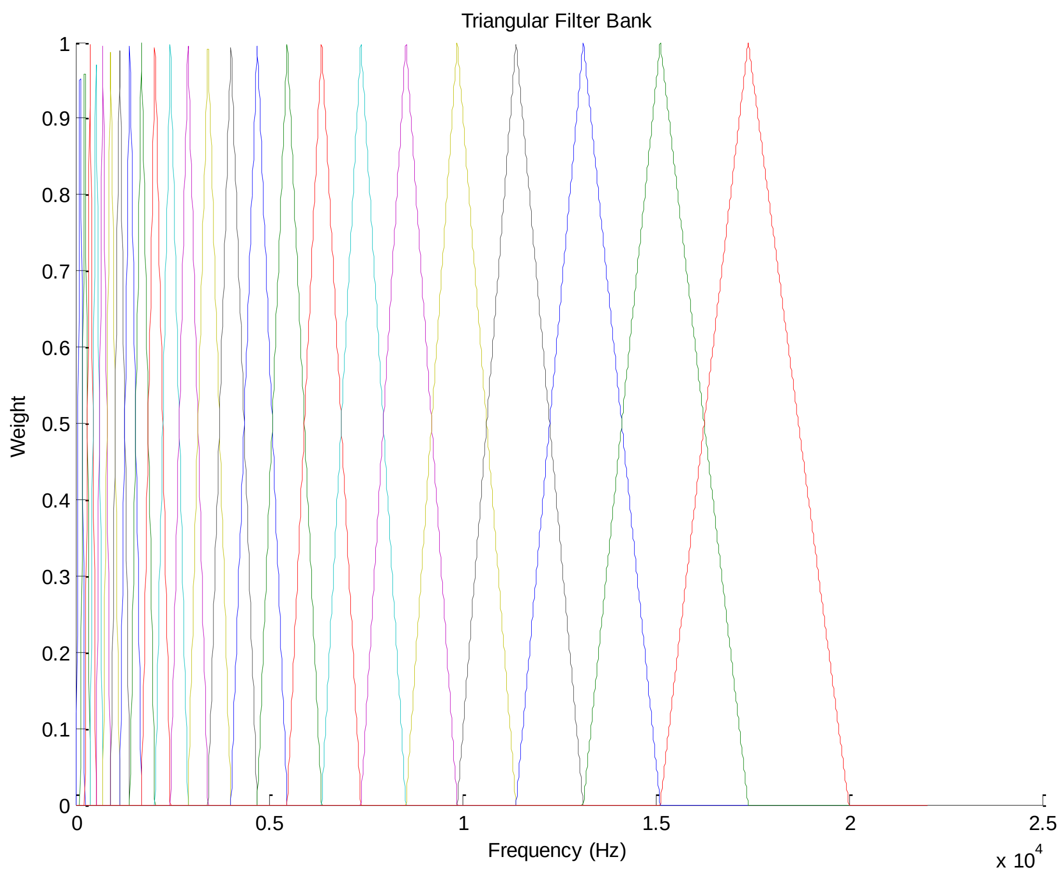

Fig 7.1: Plot for Triangular Filter Bank

7.1.2 Plot for continuous MFCC data blocks

The following graphs show the different continuous MFCC data blocks for the three genres - data_m, data_wc and data_b. These data blocks consist of MFCCs from each song of the respective genre's Training set, from rows 3 to 6 and columns 1 to 6.

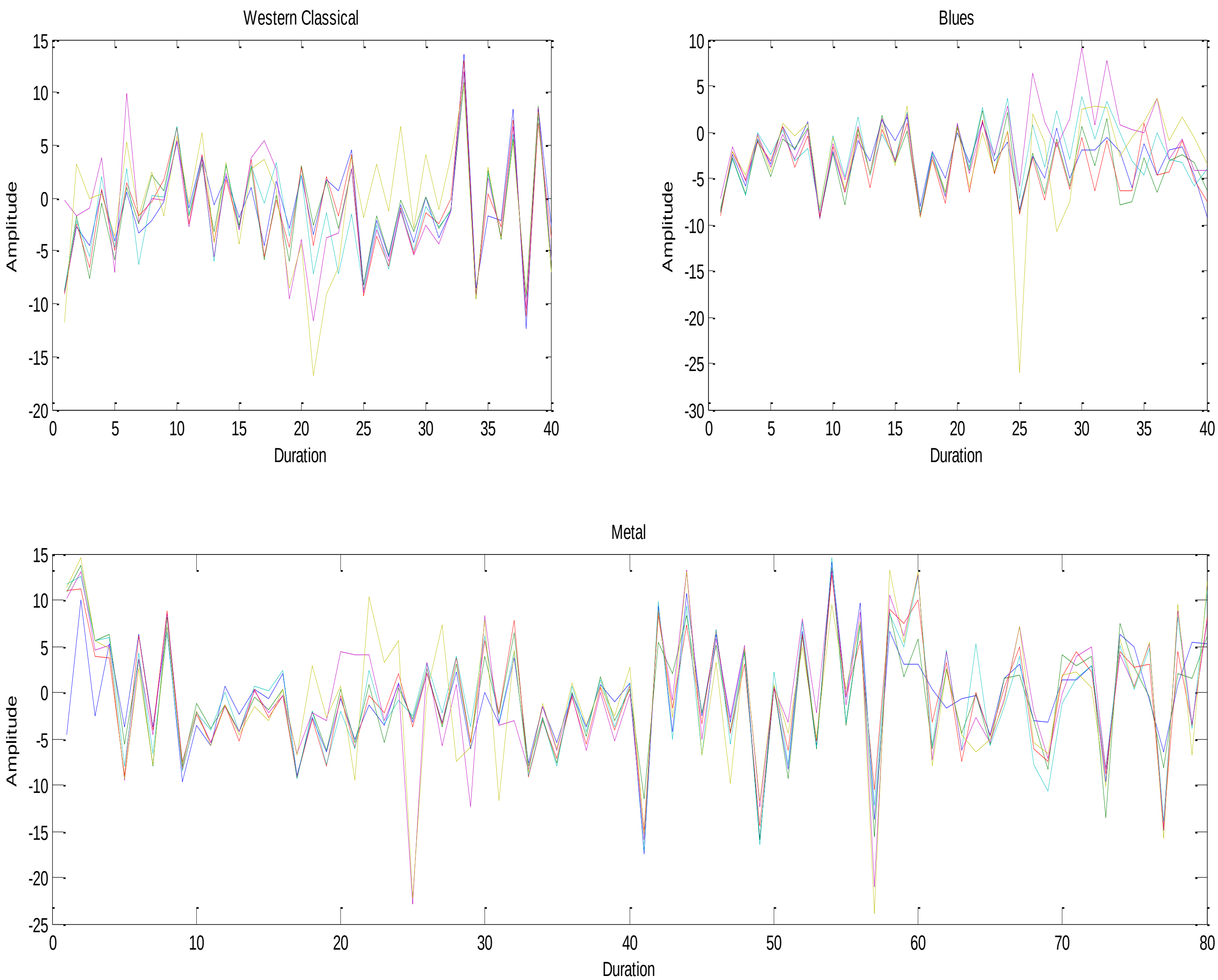

Fig 7.2: Plot for different continuous MFCC data blocks for the three genres

7.1.3 Plot for Mel-frequency Coefficients

The Mel-frequency coefficients of all three genres are potted on the same graph. It can be clearly seen that coefficients of Metal (represented in red) and Blues (represented in blue) are not overlapping with each other and coefficients of Western Classical (represented in green) overlaps both coefficients of Metal and Blues.

K Means is applied on the given plot, yielding the results given in the section 7.2 of the chapter.

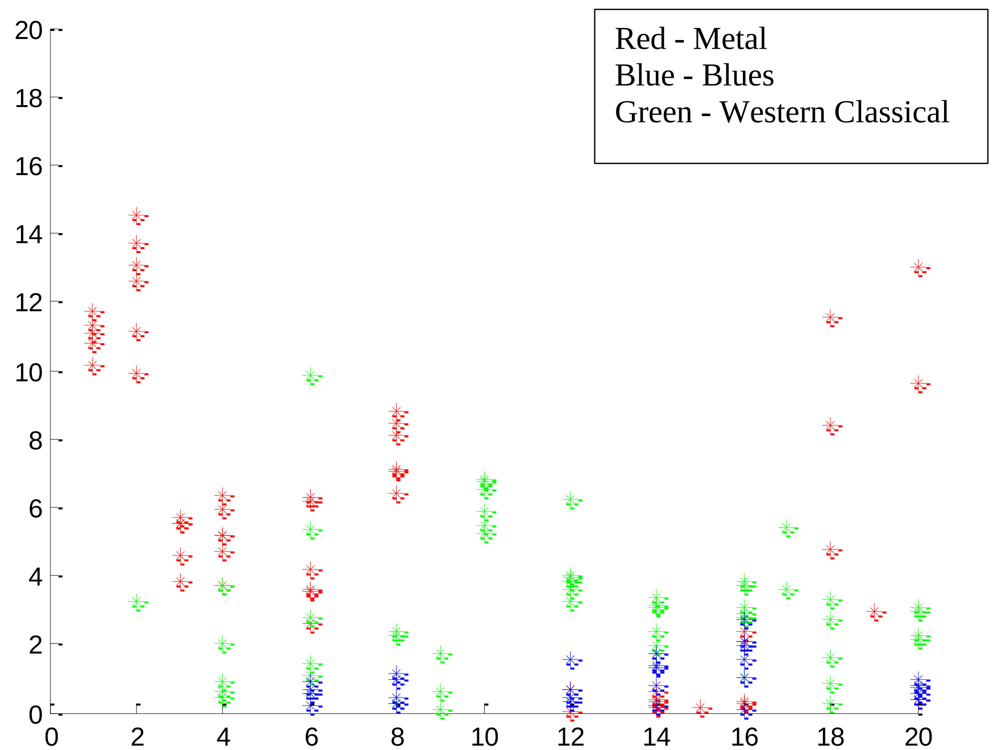

Fig 7.3: Plot for Mel-frequency Coefficients of Metal, Blues and Western Classical

7.2 K-Means

As can be noted in the picture above, the application of the unsupervised K-Means clustering algorithm has yielded positive results, with the two clusters of Metal and Blues being identified successfully, along with the detection of distinct centroids in each case. This means that distinction between a blues song and a metal song is easy when MFCCs are used, due to the extremely different timbres of the instruments used.

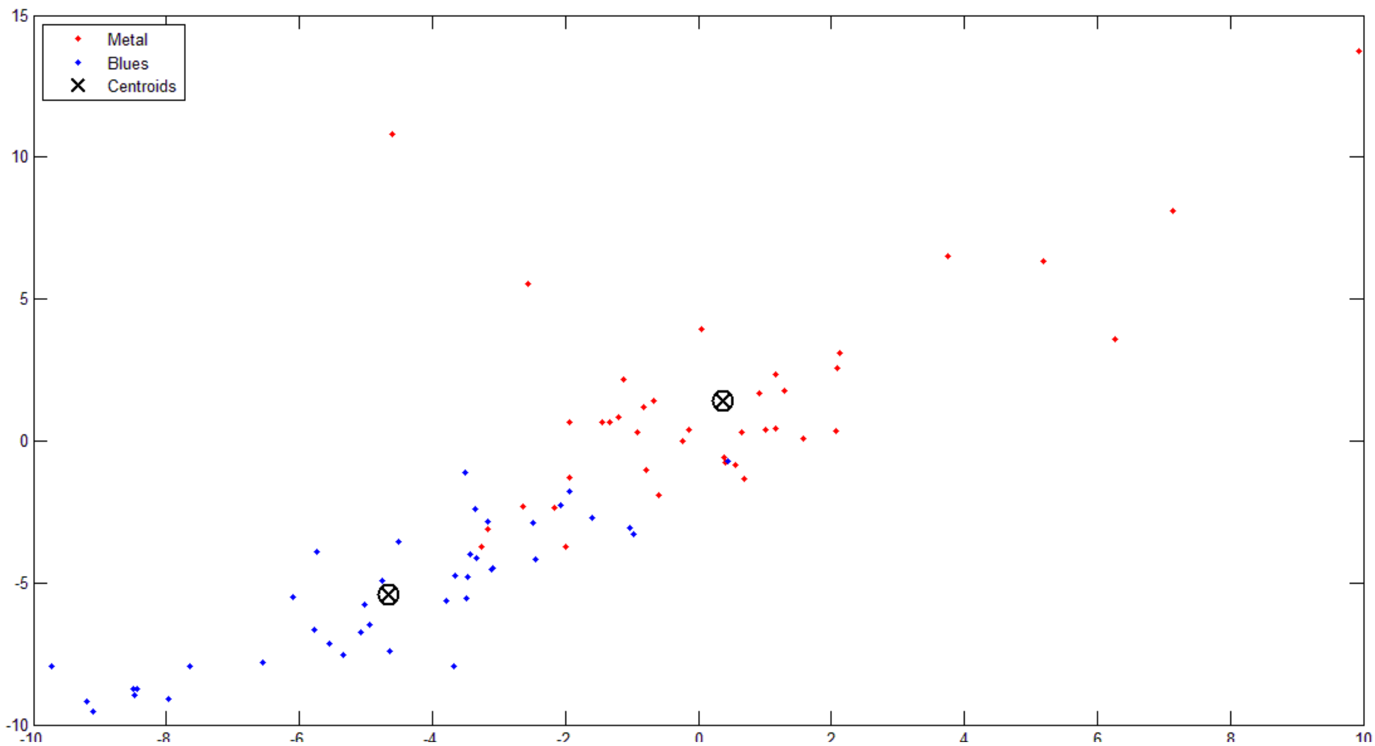

Fig 7.4: Result of K-means Clustering applied between Metal and Blues (MFCC based)

After the application of the unsupervised K-Means clustering algorithm, three distinct clusters for Metal, Western Classical and Blues were identified. These clusters have distinct centroids, which means that classification can be attempted, with decent accuracy.

However, the main issue is the fact that the K-Means algorithm is an unsupervised one, and thus cycles between the titles of Blues, Metal and Western Classical for each genre. This leads to 3 unstable results and 1 stable one. This problem can be fixed by using a supervised method.

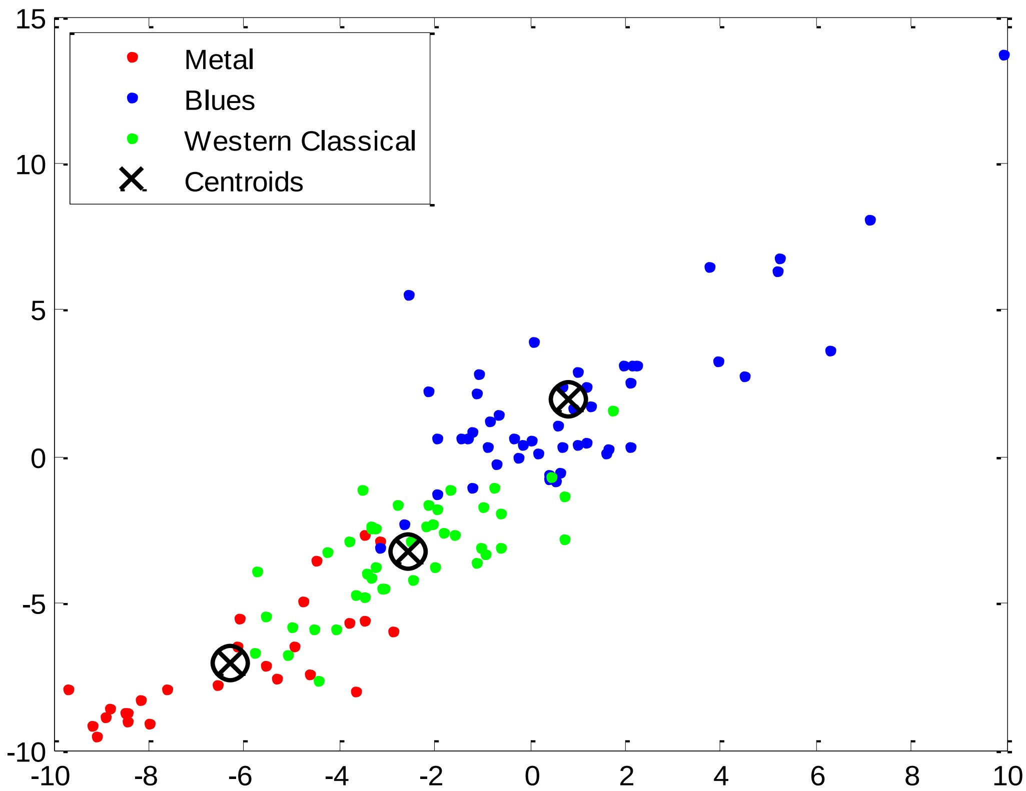

Fig 7.5: Plot showing three different clusters for Metal, Blues and Western Classical along with their centroids using K-Means

Training Set:

Table 7.1: Tabulated data showing MFCC size and K-Means cluster for the training set of the respective genres; (a) Metal, (b) Blues and (c) Western Classical

Genre	Name of the song	MFCC (size)	K-Means Cluster
Metal	N.I.B.	15x16	Cluster 1
	Paranoid	15x16	Cluster 1
	Number Of The Beast	15x16	Cluster 1
	When The Leevee Breaks	15x16	Cluster 1
	Sad But True	15x16	Cluster 1
	Fairies Wear Boots	15x16	Cluster 1
	Hard Road	15x16	Cluster 1
	Iron Man	15x16	Cluster 1
	5 Minutes Alone	15x16	Cluster 1
	By Demons Be Driven	15x16	Cluster 1
Blues	The Thrill is Gone	15x16	Cluster 2
	Ain't No Sunshine	15x16	Cluster 2
	Sittin' On Top Of The World	15x16	Cluster 2
	Poor Johnny	15x16	Cluster 2
	Mad About the Boy	15x16	Cluster 2
	How Deep Is the Ocean	15x16	Cluster 2
	Cry Me a River	15x16	Cluster 2
	Smile	15x16	Cluster 2
	The Man I Love	15x16	Cluster 2
	If I Could	15x16	Cluster 2

Western Classical	Pie Jesu	15x16	Cluster 3
	Piano Concerto No. 2	15x16	Cluster 3
	Blue Danube Waltz	15x16	Cluster 3
	Moonlight Sonata, Mvt. 1	15x16	Cluster 3
	Queen Of The Night Aria	15x16	Cluster 3
	Clarinet Quintet in A Major	15x16	Cluster 3
	Duo No.1 in G Major	15x16	Cluster 3
	Flute Quartet No.2 in G Major	15x16	Cluster 3
	Piano Sonata No.8 in A Minor	15x16	Cluster 3
	Oboe Quartet in F Major	15x16	Cluster 3

7.3 RMS

We calculate the RMS value of three genres – Metal, Blues and Western Classical. For testing, we calculate the RMS value of a test song and that value, RMS_test, is compared to the training RMS values using Euclidean distance. The RMS value to which the test value is closest belongs to the same genre. So, the song is classified.

The average RMS value a genre is calculated by taking the sum of all the training songs' (i.e., 10 sample songs per genre) RMS value and then divide that sum by 10, as shown by the following equation:

$$\text{Avg. RMS value} = \frac{\sum_{i=1}^{10} \text{RMS}_i}{10} \quad (8)$$

Training set:

Table 7.2: Tabulated results of RMS value of each song in the training set and the avg. RMS value of all the three genres.

Genre	Name of the song	RMS value $[\mathbf{RMS_i}]$	Avg. RMS value of the genre
Metal	N.I.B.	0.2385	0.1783
	Paranoid	0.1794	
	Number Of The Beast	0.1640	
	When The Leevee Breaks	0.1431	
	Sad But True	0.1554	
	Fairies Wear Boots	0.0985	
	Hard Road	0.1662	
	Iron Man	0.1821	
	5 Minutes Alone	0.2449	
	By Demons Be Driven	0.2109	
Blues	The Thrill is Gone	0.1557	0.1648
	Ain't No Sunshine	0.1693	
	Sittin' On Top Of The World	0.1251	
	Poor Johnny	0.2321	
	Mad About the Boy	0.1384	
	How Deep Is the Ocean	0.1452	
	Cry Me a River	0.1201	
	Smile	0.1622	
	The Man I Love	0.1654	
	If I Could	0.2347	

Western Classical	Pie Jesu	0.0797	0.0484
	Piano Concerto No. 2	0.0344	
	Blue Danube Waltz	0.0326	
	Moonlight Sonata, Mvt. 1	0.0323	
	Queen Of The Night Aria	0.0745	
	Clarinet Quintet in A Major	0.0714	
	Duo No.1 in G Major	0.0336	
	Flute Quartet No.2 in G Major	0.0458	
	Piano Sonata No.8 in A Minor	0.0426	
	Oboe Quartet in F Major	0.0375	

Testing Results for RMS

The accuracy of using this feature was calculated for Western Classical, for which we were able to classify 5 out of those 5 songs correctly. In the case of Blues as well, 5 out of 5 songs were classified successfully. Metal yielded the worst results of the lot, with 2 out of 5 songs being classified successfully. Softer songs were wrongly classified as Blues songs.

Table 7.3: Tabulated results for testing using RMS

Genre and it's Avg. RMS value	Name of the song	RMS value	Classified Successfully Or Not
Metal (0.1783)	Headless Cross	0.1174	**NO**
	Jerusalem	0.1470	**NO**
	Johnny Blade	0.1779	**YES**
	Juniors Eyes	0.1745	**YES**
	Kill in the Spirit World	0.1268	**NO**
Blues (0.1648)	Summertime	0.1655	**YES**
	Unchain My Heart	0.1401	**YES**
	I Put A Spell On You	0.1565	**YES**
	Hear My Train Comin'	0.1433	**YES**
	Besame Mucho	0.1678	**YES**
Western Classical (0.0484)	Piano Sonata No.16 in C Major	0.0518	**YES**
	Quintet For Piano & Wind in E	0.0469	**YES**
	Serenade No.12 in C Minor	0.0313	**YES**
	Sonata For Two Pianos in D Major	0.0618	**YES**
	String Quartet No.17 in B Flat	0.0496	**YES**

The accuracy can be calculated in the following manner:

$$\text{Accuracy} = \left[\frac{\text{No.of songs successfully classified}}{\text{Total no.of songs tested}}\right] \times 100 \quad (9)$$

Table 7.4: Accuracy results for all three genres using RMS

Western Classical		**Metal**		**Blues**	
Total no. of songs tested	No. of songs successfully classified	Total no. of songs tested	No. of songs successfully classified	Total no. of songs tested	No. of songs successfully classified
5	**5**	**5**	**2**	**5**	**5**
$\text{Accuracy} = \left[\frac{5}{5}\right] \times 100$ $= \mathbf{100}\%$		$\text{Accuracy} = \left[\frac{2}{5}\right] \times 100$ $= \mathbf{40}\%$		$\text{Accuracy} = \left[\frac{5}{5}\right] \times 100$ $= \mathbf{100}\%$	

Chapter 8

Future Scope

8.1 GUI

A graphical user interface is a user interface that allows the user to interact with the system through visual cues, like icons. The developed system could be implemented by using a graphical user interface, instead of the current text interface based implementation. This helps to overcome the steep learning curve of a command based system.

8.2 Bigger Dataset

The current dataset, though sufficient for our need, is a small representation of the wide variety of music that exists. Thus, the inclusion of more genres and songs in the dataset would result in a more robust algorithm.

8.3 More features

The current system uses two features – RMS values and MFCCs – to classify the given songs. Other features, like spectral rolloff, dynamic range and even lyrical features can be used to try and optimize the results. However, this is a double-edged path, as some features might not contribute much to the overall classification accuracy.

8.4 Different Tools

The current system utilizes MATLAB. The system could be implemented in Java, which paves the way for future integration with a internet radio platform. Tools like Marsyas, jAudio and WEKA can also be used, due to their ease of use and versatility. Additionally, features can be extracted by use of the MIR toolbox, which was not used in the project.

Appendix A

SP_12_new.pdf

ORIGINALITY REPORT

0%	0%	0%	0%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

EXCLUDE QUOTES OFF

EXCLUDE BIBLIOGRAPHY OFF

EXCLUDE MATCHES < 100 WORDS

References

1. Musical Genre Classification of Audio Signals, by George Tzanetakis and Perry Cook

2. Mel Frequency Cepstral Coefficients for Music Modeling, by Beth Logan

3. Music Genre Classification with the Million Song Dataset, by Dawen Liang, Haijie Gu and Brendan O'Connor

4. A MATLAB Toolbox to compute music similarity from audio, by Elias Pampalk

5. Automatic Classification of Music Signals, by Toni Heittola

6. Indexing Content-Based Music Similarity Models for Fast Retrieval in massive databases, by Dominik Schnitzer

7. Unsupervised Audio Feature Extraction for Music Similarity Estimation, by Jan Schluter

8. Clustering Algorithms, by Jure Leskovec and Anand Rajaraman

9. Comparisons Between Data Clustering Algorithms, by Osama Abu Abbas

10. Support Vector Machine – Concept and MATLAB Build, by Kan Xie

11. Selection of K in K-means clustering, by D.T. Pham, S.S. Dimov and C.D. Nguyen

12. The K-Means Algorithm, from Introduction to Data Mining by Tan, Steinbach, Kumar and Ghosh