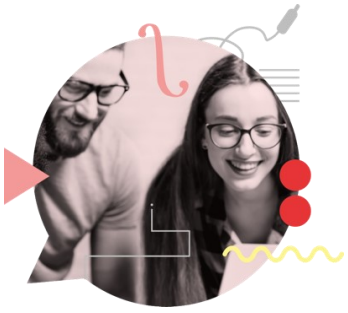
Unsupervised Classification of Sub-Genres of Electronic Music

Abhishek Choubey
Supervisor: Daniel Buner Formo



Master's program in Music, Communication, and Technology

Department of Music
Norwegian University of Science and Technology

Department of Musicology
University of Oslo

May 2022

Abstract

This thesis is a study of unsupervised machine learning techniques for the classification and clustering of sub-genres of electronic music. New sub-genres of electronic music are frequently introduced and most have similar audio characteristics, having a proper distinction between them is a laborious task. Therefore, it becomes essential to explore tools and techniques that help us differentiate between these genres easily and efficiently. Two approaches suggested by Barreira and Rauber have been employed for the clustering of music. Barreira's approach uses a model-based clustering technique by employing Expectation-Maximization for Gaussian Mixture Models. Whereas, the Rauber approach uses Growing Hierarchical Self Organizing Maps which is an extension of Self Organizing Maps. Moreover, Low-level audio features that mathematically show characteristics of audio are extracted for feeding into these algorithms. The thesis is concluded by reflecting upon the results, evaluating the models, discussing limitations, and proposing future works.

Acknowledgment

First of all, I would like to thank my supervisor Daniel Buner Formo for his continuous support, and guidance and for providing me with a better direction for the development and implementation of the system and the writing processes. Your propensity towards minor details and clear direction has been very helpful in the completion of this project.

A special thanks to Stefano Fasciani and Øystein Fjeldbo for being amazing teachers and great mentors during the two years of the MCT program. Stefano's dedication to the course and his vision is very motivating. Øystein's support and friendliness have been very heartening.

I would also like to thank my classmates; your collaborative and fun personas made these two years to remember forever.

Lastly, I would like to thank my mom, Abha Choubey, and my sister, Aditi Choubey for providing constant support and belief, thank you for being there every step of the way.

Contents

1 Introduction 1
1.1 Research question and motivation 2
1.2 Contribution 3
1.3 System overview 4
2 Background 6
2.1 Music Information Retrieval 6
2.1.1 Importance of MIR 7
2.1.2 MIR and music analysis 7
2.1.3 Previous and current research on music genre classification and its basis. 8
2.2 Audio signal processing and feature extraction 9
2.2.1 Audio signal characteristics 10
2.2.2 Digital conversion of the analog audio signal 12
2.2.3 Audio signal features and their extraction 13
2.3 Machine learning and music 17
2.4 Summary 19
3 Methods 20
3.1 Choosing the systems 21
3.2 Database selection 22
3.3 Evaluation techniques 22
4 System description and Implementation 25
4.1 Data acquiring and pre-processing 25
4.2 Feature extraction 26
4.2.1 Spectral Centroid (SC) 27
4.2.2 Spectral roll-off 27
4.2.3 Spectral flux: 27
4.2.4 Zero-crossing rate (ZCR): 28
4.2.5 Mel frequency cepstral coefficients (MFCC) 28
4.2.6 Root mean square (RMS) energy 29
4.2.7 Bandwidth 30
4.2.8 Power spectral density 30

4.3 Barreira system description 31
4.3.1 Feature matrix and Standardization 31
4.3.2 Dimensionality reduction using Principal Component Analysis (PCA) 32
4.3.3 The Clustering stage 32
4.4 Rauber system description 35
4.4.1 Self Organizing Maps (SOM) 35
4.4.2 Growing Hierarchical Self Organizing Maps 38
4.4.3 The Clustering stage 40
5 Experiments and Results 41
5.1 Barreira's model-based approach 42
5.2 Rauber's Self Organizing Maps and the Growing hierarchical Self Organizing Maps 43
6 Conclusions 46
6.1 Limitations and Future Work 47
References 49
Appendix 59

1 Introduction

Music Information retrieval (MIR) is an interdisciplinary field concerned with the development of innovative approaches to streamline the plethora of digital music and provide easy accessibility by extracting features from music (audio signal or noted music) and by developing different search and retrieval schemes (Schedl et al., 2014). Given the importance of music in our society, it's surprising that MIR's research is still relatively new, having widely started around two decades ago (Burgoyne et al., 2015), MIR has however undergone a transition since then. As a scientific field, it has been steadily improving and some of the most crucial reasons for its success are (i) the development of audio compression techniques in the late 1990s, (ii) increment in the computational prowess of the personal computer which in turn resulted into extraction of information by users in a reasonable amount of time, (iii) widespread availability of music databases and more recently (iv) the emergence of music streaming services like Spotify, Pandora, iTunes, etc. As the number of digitalized music being uploaded on the internet keeps increasing, consumption of the music is also increasing as well as the number of styles, genres, and themes. With this comes the challenge of differentiating between these genres and styles. A number of different approaches have been proposed under the umbrella of Music Information retrieval for the same. Originally introduced as a pattern recognization task by (Tzanetakis & Cook, 2002) music genre recognization and classification have become a prominently needed tool for easy retrieval of music in search engines and music databases. According to a study by (Aucouturier & Pachet, 2003), music genre classification is one of the most common ways used in managing musical databases. On one hand, music may be classified into one or more musical genres, but cultural differences and human perceptions make establishing a standard music genre taxonomy problematic. An avid listener well familiar with electronic music, for example, could categorize between a minimal techno track with a melodic house track but for non-listeners, they may all fall into techno, house or even just electronic music. The "similarity" in music is difficult to describe and particularly complicated due to the numerous elements (timbre, melody, rhythm, harmony), which are among the aspects that define the viewpoint of music. And it becomes increasingly laborious to distinguish

between music genres if they are similar to each other or if they fall under a particular genre of music.

This thesis aims to cluster the sub-genres of electronic music that have similarities in their musical elements by implementing two approaches proposed by (Barreira et al., 2011) and (Rauber et al., 2002). These approaches have been proposed for clustering and classification of popular genres like pop, hip-hop, classical, rock, etc. On the contrary, this thesis aims to implement them in sub-genres of electronic music which have a considerable amount of similarity between them. The thesis begins with an introduction, clarifying the research question and motivation, followed by a brief description of the approaches. Secondly, the background is explored. Following the methods for choosing the algorithms and the evaluation techniques are explained. Then it details the feature extraction pipeline and the features to be extracted as proposed in the approaches by Barreira and Rauber, followed by a detailed description of the system and the techniques employed. Finally, the evaluation results of the applied approaches are discussed. The thesis is concluded with a reflection on limitations, future work, different scopes, and the influence of the research project.

1.1 Research question and motivation

Several clustering and classification techniques have been introduced and implemented for the classification of genres of music. Ranging from a number of approaches using supervised machine learning techniques (Ahmad et al., 2014; Asim & Ahmed, 2017; Bahuleyan, 2018; Tzanetakis & Cook, 2002), etc. In these techniques, the model is trained on input data that has been labeled in correspondence to a particular output. Moreover, some employ semi-supervised classification (Poria et al., 2013; Song & Zhang, 2008) approaches which are a combination of supervised and unsupervised machine learning approaches. However, these techniques necessitate manually tagged data. As a result, these approaches are restricted in their ability to evolve with new data, music, and genres, as manually constructing and updating a big dataset for machine learning models is not always viable. The unsupervised approach is a branch of machine learning in which the model is not given any labeled data to train itself, rather it autonomously works to find patterns and information.

Additionally, it is not constrained since it does not require musically labeled data to work and is thus autonomous in classifying musical genres based on sample audio attributes, as it takes into account the features related to the data. However, just a few solutions have been presented, ranging from using hierarchical self-organizing maps(Ahmad et al., 2014), employing the Hidden Markov model (Barreira et al., 2011), or more recently using an unsupervised artificial neural network(Pelchat & Gelowitz, 2020) (Raval, 2021). Moreover, these solutions majorly explore the classification of relatively easily distinguishable genres which are widely popular like pop, hip-hop, rock, classical, etc. The elements that differentiate between them are very distinct from each other and have very less overlapping. The classification of genres that are similar to each other or have similar elements present in them has very minimal exploration, like (Quinto et al., 2017) wherein classification of subgenres of Jazz music using deep learning techniques has been performed.

Motivated by the concepts of unsupervised machine learning, audio signal processing, and feature extraction techniques, as well as identifying the lack of, and thus the need to do research exploration of classification of similar music type leading to the classification of sub-genres of electronic music my research question becomes: How to efficiently implement unsupervised machine learning techniques that cluster or classify mainstream genres to different sub-genres of electronic music? and subsequently enhance and evaluate them. By reflecting upon this question my objectives were realized accordingly. Firstly, implementing the algorithm proposed by senior researchers in the field that are proven successful and applying them to the particulars of my research question, secondly choose the most optimum elements (types of algorithm, types of features, and evaluation criteria) from the proposed approaches in reference to my research question.

1.2 Contribution

The primary goal of this thesis has been to explore unsupervised machine learning techniques for the classification and clustering of musical genres that have some similarities to them. There are several unsupervised approaches that have been implemented and proposed for various applications but to narrow the scope of the thesis unsupervised approaches presented by (Barreira et al., 2011; Rauber et al., 2002) are applied. This study aims to fill the research gap in the field of MIR for music genre

classification using unsupervised machine learning techniques. It also aims to explore and bolster the features and feature extraction techniques presented by the researchers mentioned above and evaluate them on different styles of music. Subsequently, the study offers a small contribution to the understanding of sub-genres of electronic music, what makes them different or common to each other, and their impact on MIR as well as societal influence. It also explores the possible ways by which the classification between them provides merit to the MIR society and music consumers. Additionally, It's also worth noting that several approaches to the problem of music genre classification have been proposed in the MIR research timeline. They are based on a variety of factors associated with music, such as metadata, the artist or composer, the historical period in which the musical composition was created, or the geographical area from which the music originated, as well as cultural references and connotations. However, to limit the scope of this short-term one-semester thesis project, only two types of approaches are considered which are based on low-level audio features and widely popular machine learning algorithms. In addition to that, only the clustering of data provided to the algorithm is performed and no new data has been introduced after the models have been generated and implemented.

By the completion of the thesis project, it aims to provide a valuable implementation and assessment of computational tools depicted by the researchers Barreira and Rauber for the classification of musical genres and that it succeeds in generating a deeper understanding of the complete system of machine learning including audio signal processing and other MIR fields.

1.3 System overview

There are two separate systems that are being implemented in this thesis. The first system as proposed by (Barreira et al., 2011) uses a model-based approach for the classification of genres. In this approach, the clustering method consists of a learning approach that clusters the music samples only based on their audio features without any previous information about the genre of the samples. The first step in this approach is to extract features, these features are related to the properties of the audio sample such as spectral analysis, timbre, loudness, and melody among others. These features when extracted have a large number of dimensions and therefore a feature reduction technique is required for optimizing computational time, storage space, and redundancy. After standardization

of the features dimensionality reduction technique based on Principal Component Analysis (PCA) (Abdi & Williams, 2010) is used. After the feature reduction step, Model-Based Clustering Analysis (MBCA) as proposed by Fraley and Raftery (Fraley & Raftery, 1998) is implemented in which the classification of the music samples is done using EM (expectation-maximization) algorithm for maximum likelihood, with Gaussian Mixture Models, which essentially is a clustering technique in which each sample is given a probabilistic assignment to the cluster it may belong to and then is joined with the samples which are closest in similarity to the original.

The second system uses psychoacoustic models and self-organizing maps for classification purposes and is proposed by (Rauber et al., 2002). In this approach first, the low-level feature from the audio signal is extracted which is based on the frequency spectrum. After this, the next step is to provide this feature to a neural network, defined by Rauber as Growing Hierarchical self-organizing maps(GHSOM) which is essentially an extension of Self-organizing maps(SOM). By placing related data items next to one other on a map display, this neural network does cluster analysis. The GHSOM, in particular, is capable of recognizing hierarchical correlations in data and so generates a hierarchy of maps depicting distinct musical genres into which the pieces of music are arranged (Rauber et al., 2002).

2 Background

This chapter delineates the previous work and theoretical background linked to the system and algorithm given in this thesis. First, an introduction and historical view of the Music Information retrieval field is discussed with a focus on music genre classification, and electronic music. Then, previous and current research in the niche topic of music genre classification is explored Subsequently, the audio signal processing field with a focus on feature extraction is discussed. After that, a brief description of machine learning and how it relates to music and the context of this thesis project is given.

2.1 Music Information Retrieval

Music is and has been a central topic in our society, it has ignited and transformed cultures and behaviors, and almost everyone enjoys listening to music and many relish creating it as well. Broadly speaking, the interdisciplinary research field of Music Information retrieval (MIR) is mainly concerned with developing novel approaches to ease the access to a large amount of music present around us, by feature extraction and inference of relevant information from either the audio signal or symbolic representation or from external sources like webpages or even from a giant library of metadata which includes information about artists, genres, culture, etc as defined by Downie (Downie, 2004). As a consequence MIR has facilitated access to music for everyone by lowering the barriers. MIR is responsible for innovations such as customized music suggestions, software that assesses the key and speed of tracks to aid DJ mixing, scanners that turn printed music into electronically modifiable sheets, and a variety of additional digital interfaces to musical information. The significance of MIR will only increase as more people engage with music online (Downie, 2004).

2.1.1 Importance of MIR

One of the prominent fields in which the concept of MIR is applied is the very popular music genre classification. Its wide popularity can be understood in the sense, as the number of huge databases is arising either because of the restoration of analog archives in digital format or the addition of new content every day it becomes increasingly tough to cater to all the new content. As a result, a need for reliable and fast tools to organize these datasets becomes imminent. In this context, music genres are crucial descriptors since they have been used to organize music in catalogs, libraries, music stores, etc for a long time. Despite their widespread use, music genres remain a vaguely defined notion, making automated categorization a difficult undertaking. Music genres are styles of music and categories that have emerged as a result of a complex interaction of cultures, artists, geographies, and commercial pressures. As described in the survey by (Scaringella et al., 2006), we can see how different the requirements that identify a single genre might be, by looking at some distinct and extensively utilized music genres. For example:

- Indian music is geographically defined.
- Baroque music is related to an era in history while encompassing a wide range of styles and a wide geographic region.
- Barbershop music is defined by a set of precise technical requirements.
- Electronic music is devised by music made using computers which can further be classified into several sub-genres.

Therefore it becomes important to develop approaches that can help in the taxonomy of these vast and not so concisely defined descriptors of styles of music.

2.1.2 MIR and music analysis

The research in the field of music analysis however started quite early, even before the age of computers to find meaningful information in different musical styles. Early research in MIR predominantly focused on working with the symbolic representation of music, with development in modern statistical methods some scholars and musicians were applying it to the music which can be spotted in the work done as early as the beginning of the 20th century. Researchers like Myers published in 1907 that *larger melodic intervals occur less frequently in folk music than smaller melodic intervals.* As the computer became more widely available and accessible to researchers in the mid-20th century, the interest in music computational

analysis grew and the terms like computational musicology and Music Information Retrieval were introduced in the papers published by (Kassler, 1966; Logemann, 1967). After some early research in pitch tracking, tempo estimation, and timbre analysis (Moorer, 1975; Slawson, 1968), MIR later saw a decline in research opportunities as it became harder to access a large amount of digital data due to lack of and also the laborious task of gathering pertinent data to analyze with limited computational prowess. In the 1990s, as the computational power grew MIR saw a rise again, as along with this, digitalized musical data also became available to a large extent. Unique research largely confined to the field of MIR started popping up, the trendsetter was the ever-popular query-by-humming research by (Kageyama, 1993), which appeared in the first half of the decade, followed by studies on using audio material to search databases (Wold et al., 1996).

2.1.3 Previous and current research on music genre classification and its basis.

The music genre problem asks for a taxonomy, meaning a structured set of divisions that can be overlaid onto a music collection. (Daniel & Cazaly, 2000) examined a variety of music genre taxonomies used in industries and on the internet, demonstrating that constructing such a hierarchy of genres is difficult, as music genres are not very accurately defined. Moreover, one basic question that can be raised while classification is on what basis or element of the music should a classification be built on? Can it be the title of the song, the album, or maybe even the artist? If we place one song into only one kind of genre, it doesn’t seem to work as some songs have elements of different genres. Furthermore, an album can be a mix of heterogenous styles or genres of music. Similarly, an artist can make different kinds of styles of music which can fall into various genres, so limiting a music piece to a genre because one particular artist made it, is simply not practical. While some of this metadata can work, they are not always universal or reliable. One thing to note here is that some high-level representative features of music also work exceptionally more or less accurately as compared to low-level features discussed below, typically they are, event-like formats such as MIDI or symbolic formats such as MusicXML, and many researchers have exploited MIDI data in particular as the basis of music analysis and genre classification.

Even so, exploiting the audio signal's content is a far superior technique., as it can offer significant information about the music in terms of timbre, pitch, tempo, instruments used, and many more. So it becomes natural to use these audio contents as the basis of classification. While using a small

timeframe of a sample from a full-length track seems to work it is again not the best method since there is a very low level and density of information in them. Moreover, a combination of a number of these audio samples generates a huge amount of data to be processed. With all of these factors in mind, demand for low-level features emerges. Based on these factors, research on music genre classification in its nascent stages was explored when it was first introduced as a pattern recognization task by (Tzanetakis & Cook, 2002). Later, according to a study by (Aucouturier & Pachet, 2003), music genre classification was depicted as one of the most common ways used in managing musical databases. Numerous works have been carried out to understand the similarity in music and then classify them, ranging from comparing metadata of the songs (Pérez-Sancho et al., 2009) to understanding audio features of the songs (Asim & Ahmed, 2017) (Costa et al., 2012) (Keum & Lee, 2006), or by using MIDI data (Cataltepe et al., 2007) as the basis of classification. The research studies that employed audio content as the foundation of analysis were the most successful (Cataltepe et al., 2007; Costa et al., 2012; Pérez-Sancho et al., 2009; Quinto et al., 2017; Raval, 2021; Wold et al., 1996)

2.2 Audio signal processing and feature extraction

The hearing sense provides us with rich information about our surroundings and environment concerning the location and characteristics of the sound and the sound-producing objects. For example, we can effortlessly absorb multiple sounds present in a sonic field like sounds of birds chirping, to the traffic noise, while listening to a song on the speakers. By analyzing and categorizing measurable sensory inputs, the human auditory system is able to comprehend the complex sound mixture hitting our ears and generate high-level conceptions of the world. This is also called auditory scene analysis which is the process of separating and identifying sources from a composite audio signal that has been received (Rao, 2008). It is very easy to understand that if this property is implemented in a machine it will be very useful in tasks like speech recognition, instrument recognition, and other MIR fields. Some important applications of audio processing are audio compression, audio synthesis, and audio classification. In the recent past, audio compression has been the dominant field in the field of audio signal processing with research papers emerging around the 1980s, authors from numerous research and development laboratories, including Erlangen-

Nuremberg University and Fraunhofer IIS, AT&TBell Laboratories, and Dolby Laboratories, began presenting audio compression research papers at IEEE ICASSP and Audio Engineering Society conferences. Although most audio compression algorithms were designed as part of digital motion video compression standards, such as the MPEG series, they later became essential as stand-alone audio recording and playback technologies. Progress in VLSI technology, psychoacoustics, and efficient time-frequency signal representations enabled the development of a series of configurable real-time compression algorithms for audio and film applications (Spanias et al., 2006). In modern times the increasing importance of digital media and its management has accelerated growth in the technologies related to audio segmentation and classification. Audio classification however is a subset of a larger issue of management of audiovisiual data. Speech and speaker recognition are classic audio retrieval problems that have gotten decades of study attention. On the other hand, the fast-expanding online archives of digital music are bringing attention to broader issues of nonlinear browsing and retrieval, utilizing more natural methods of engaging with multimedia material, most notably music. Because audio records (unlike photos) may only be listened to in order, excellent indexing is essential for efficient retrieval. Listening to audio bits rather than watching video sequences can actually make it easier to traverse audiovisual content (Rao, 2008). Therefore within the context of this thesis and generally, it becomes essentially important to research and develop approaches for audio signal processing, classification, and retrieval.

2.2.1 Audio signal characteristics

Before we could develop or implement audio classification techniques, it's natural to first understand the properties of an audio signal, what it consists of and how can we use that information to extract features that will be meaningful for our objective of classification. The human auditory system is responsive to the sound laying between the frequencies of 20Hz to 20kHz, and so we mostly deal within this range for our audio application purposes. Figure 2.1 shows how the human auditory system responds to sounds in the 20 Hz–20 kHz range. It's a graph that shows the relationship between sound pressure level (SPL) in decibels and audible frequency range (Miller, 1951; Sharma et al., 2020). The graph shows the absolute threshold of hearing for different types of sounds. For capturing the sound and reproducing it, a microphone speaker pair is ubiquitous. The sound captured by the microphone is a time waveform of the variation of the air pressure around it in the sonic field in which the microphone is present.

The electrical output of the microphone is sampled and quantized appropriately to produce a digital audio signal. Although any sampling frequency over 40 kHz would be sufficient to record the complete range of audible frequencies according to the Nyquist theorem (Landau, 1967), 44,100 Hz is a popular sampling rate that originated from the necessity to coordinate audio with visual data in the past. 44.1 kHz sampled audio converted to 16-bit length is called "CD quality." We can broadly categorize sound into the following categories: speech, music, environmental sounds, and artificial sounds. Most of the interesting sounds change with time, that is, they are time-varying sounds and are made of either group of impulse sounds or evolving patterns. All of the attributes of a sound can be described by a combination of its temporal properties and spectral properties. The temporal properties of a sound can be described as the properties related to the duration, intensity, and amplitude changes of the sounds. Whereas spectral properties can be defined as the ones that are related to the frequency of the sound and its strength. These properties are obtained by converting the time-based signal into frequency based using fourier transform. Audio waveforms as mentioned earlier can be periodic or non-periodic. Periodic audio waveforms, with the exception of the basic sinusoid, are complex tones comprised of a fundamental frequency and a succession of overtones or multiples of the fundamental frequency. The color or the “timbre” of the sound is the result of the relation between the amplitude and phases of these frequency components. The aperiodic waveform on the other hand generally consists of non-harmonically linked sine tones and noise frequencies. Generally, the quality of the sound depends on the combination of these noise-like and tone-like frequencies (Rao, 2008). For example, frequencies in a musical piece since it have a melodic sequence of notes that are highly tonal for the most part with both fundamental frequencies and their variation varying over a wide range. Whereas speech on the other hand consists of tonal and noisy regions. Sound analysis by the human auditory system is carried out by frequency analysis of the sound to give inputs to our brain to perform higher cognitive tasks. The analysis compasses of evaluation of both the spectral and temporal features of the sound, as both of these features, are important for the perception and cognition of the sound (Moore & Linthicum, 2007). So, it is natural to represent the audio signals through a joint addition of time and frequency.

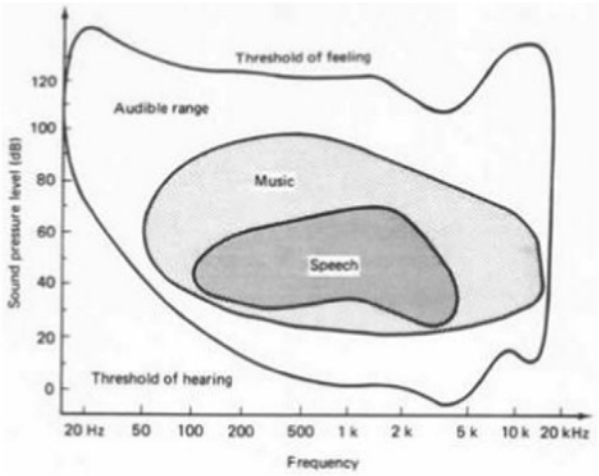


Figure 2.1: Absolute threshold of hearing (Sharma et al., 2020)

2.2.2 Digital conversion of the analog audio signal

Since an audio waveform is a mechanical wave created by the energy difference of the environment around the source, it is analog in nature. To process and analyze this waveform computationally it should be converted into a digital format. While analog signals are continuous in nature and when plotted can theoretically have an infinite resolution in terms of the amplitude and time of the signal as they are plotted on a real number axis. It is not the same with a digital signal, a very high resolution is not practical as it requires a huge amount of data storage. Therefore we need to convert an analog signal into a digital signal so that we can use it to manipulate or extract features and information about the sound. This conversion process also called analog to digital conversion(ADC) consists of two sub-steps: Sampling and Quantization. Sampling is nothing but acquiring of data points across an analog wave or a sound wave at specific points in time. A sampling of these data points is done on equidistant points in time and the distance between these points is called the sampling period and is denoted by T, consequently, the inverse of this period is called the sampling rate sr, which gives us the information about the number of samples present in one second. Now after we have allocated these data points the next step called quantization is implemented. Quantization can be described as acquiring data points similarly to sampling but instead for the amplitude properties of the waveform. In this process, we have a fixed discrete number of amplitude values on the y-axis and then at each sample, we just quantize the value of amplitude to the closest discrete number, and we reproduce the analog sound waveform in a digital format. It is intuitive to understand

that the higher the quantization value and the sampling rate the higher the resolution of the digital waveform will be.

2.2.3 Audio signal features and their extraction

Audio features are descriptors of sound which provide us with different information about the sound. The goal of these features is to be compact and small in size in comparison to the audio signal and still highlight the characteristics of the signal. The features are chosen in such a way that they considerably reduce the size of a signal while still properly characterizing it. The simplified form of the signal is especially important in the context of this thesis as it reduces the computational and temporal complexity of ML algorithms, making them more appropriate for real-time applications. So, feature extraction is a signal dimension reduction procedure that makes the signal more amenable for machine learning algorithms (Sharma et al., 2020). Audio features as described by (Knees & Schedl, 2016) can be broadly categorized into 4 high-level categories:

- Their Level of extraction
- Their Temporal scope
- The Musical aspect they describe
- The Signal domain they are computed in

The *level of extraction* of a feature is usually defined on a 3-level scale: Low-level. Mid-level and High-level. As we go from low level to high level the conceptual meaning of the feature to the user decreases and the closeness of the feature to the raw waveform increases. Low-level features are generally straightforward statistical summaries of the waveform derived directly from the raw audio waveform. At the highest level features are described on the basis of human perception. While low-level features can be features like amplitude envelop, energy, zero crossing rates, spectral flux, etc, mid-level features are pitch and beat related descriptors, note onset, fluctuation patterns, etc, and high-level features are features like instruments, chords, melody, tempo, rhythm, etc.

Features based on *their temporal scope* can be distinguished as instantaneous, segment level, and global level features. Instantaneous features are calculated at a particular point in time, Since the temporal resolution of the human ear is around 10ms for most healthy persons (Madden & Feth, 2018), instantaneous features are mostly calculated in the range of at most a few tens of milliseconds. Segment level features are the ones that are calculated over a segment of audio and can range from 5 seconds to sometimes up to 20 seconds, these features are the ones that

give us information about say a bar or a musical phrase. Global features describe the entirety of the audio or music sample in question like an entire song or an audio excerpt

The features related to the *musical aspect* intuitively describe the musical characteristics of the sound, such as beats, rhythm, tempo, melody, pitch, timbre, instrumentation, etc

Another way to classify music characteristics is by the *signal domain* in that they are calculated. An audio signal can be depicted in the frequency domain or in the time domain. The time-domain features indicate the amplitude of a signal at each point in time or in other words at each point in time when the sample was taken. The frequency-domain features are basically the product of the Fourier transform of the signal and give us information about the magnitude of the frequencies present in the signal. For an illustration of the signals in these two domains, consider the following figure 2.2.

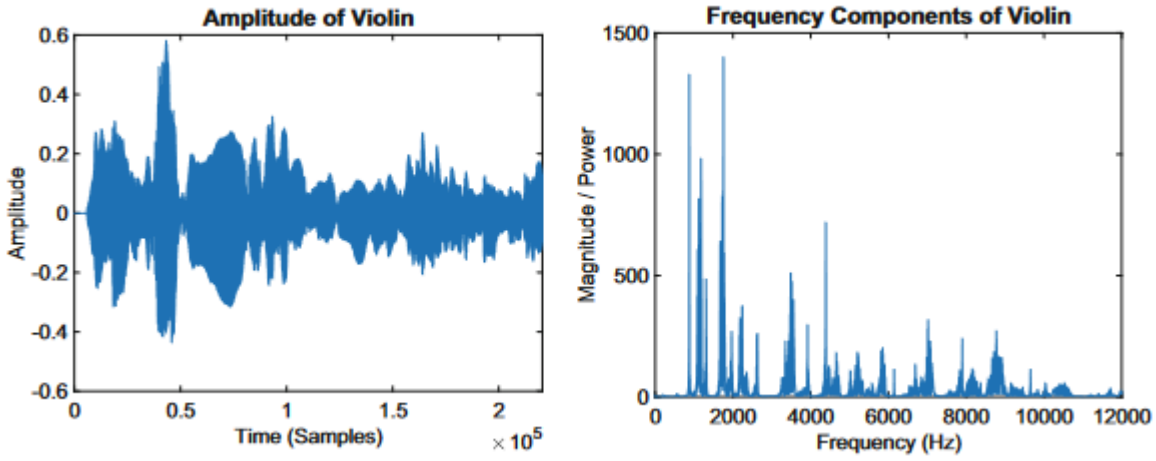


Figure 2.2: Time-domain representation (left) and frequency domain representation (right) (Knees & Schedl, 2016)

The widespread research of audio features began in the 1950s, when communication engineers began investigating speech analysis, first focusing on time-domain features (Goldman-Eisler, 1958; Miller, 1951; Stevens, 1950). Researchers started concentrating on frequency domain features after the time domain features arose until the late 1950s. Time-domain elements have always been significant in audio analysis and categorization. Pitch, formants, and other frequency domain properties have been developed and used in diverse applications to assess the

spectrum of audio signals even in modern research and commercial projects. Later in the 1960s, the combined time-frequency features extraction techniques started popping up (Gambardella, 1968, 1971; Rihaczek, 1968) and have been employed in audio signal processing tasks since then. In the recent past, deep features have been applied in audio signal processing in the areas of acoustic scene categorization speaker recognition, and audio-video analysis as well as other applications since the advent of deep learning (Li, Zhang, et al., 2018, 2018; Quinto et al., 2017).

Even though different techniques have been developed and evolved over time, the underlying architecture for most of these techniques remains the same. To compute some kind of acoustical audio feature a basic pipeline has been implemented as shown in figure 2.3. The process starts with capturing the sound from the source using a microphone and converting the analog signal into a digital signal, also called analog to digital conversion (ADC) using sampling and quantization. As a consequence, the audio is represented using so-called pulse code modulation (PCM). In this process, the amplitude value is associated with each sample measured at a set time interval. Because these samples represent a single moment in time, alone they are too short to process and get meaningful information from, it can be understood if we take commonly used values for digital conversion of the audio signal. If a single sample is considered at a rate of 44.1kHz and the time duration of a sample is given by, where T is the time period and sr is the sampling rate

$$T = \frac{1}{sr}$$

Then the duration of a single sample comes to be around 0.0227ms which is very low than the threshold of human perception of sound i.e, 10ms. In order to overcome this, we start by concatenating these samples into frames. Frames are nothing but audio chunks that are long enough to be perceivable. The duration of a frame is given by

$$D_f = \frac{1}{sr} \times K$$

where D_f is the duration of a frame sr is the sampling rate and K is the frame size (no of samples in a frame)

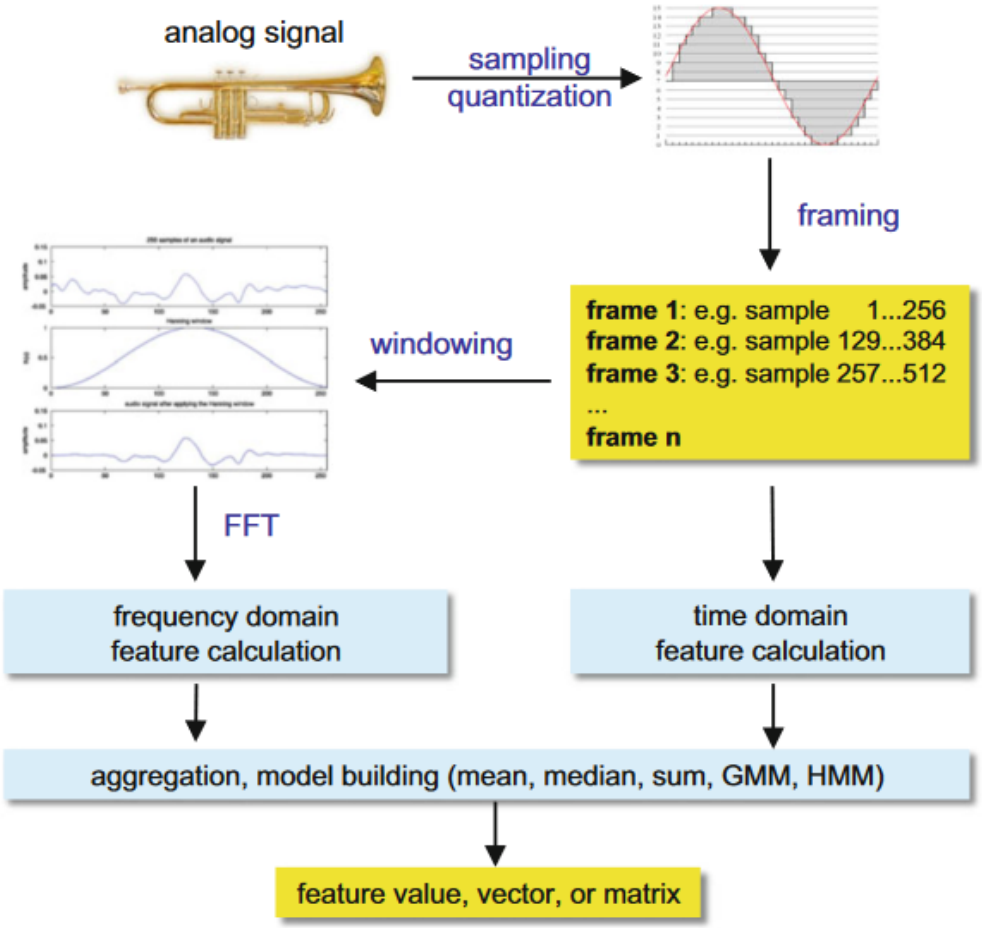


Figure 2.3: Simplified representation of a feature extraction pipeline as shown by (Knees & Schedl, 2016)

The entirety of the audio signal is thus represented by a number of frames. Using this frame representation, it is already possible to calculate time-domain features. However, for computing frequency domain features an additional step called windowing is essential. Windowing is fundamentally applying a window function to each frame of the audio signal. This is done to reduce the problem called spectral leakage. If the starting and ending of a given frame are not periodic the Fourier transform of that frame results in the formation of artifacts, which are visible in the higher frequency region which is also called spectral leakage. To avoid this a very widely used windowing function called the Hann or Hanning function is implemented, which eliminates samples at each end of the frame making it periodic, and is given as

$$w(k) = 0.5\left(1 - \cos\left(\frac{2\pi k}{K-1}\right)\right), k = 1 \dots K$$

Here small k is sample, K is the number of samples in a frame. Moreover the windowed signal $s_w(k)$ is given as

$$s_w(k) = s_k \times w(k), k = 1 \dots K$$

Here, s_k is the signal and $w(k)$ is the Hann window function.

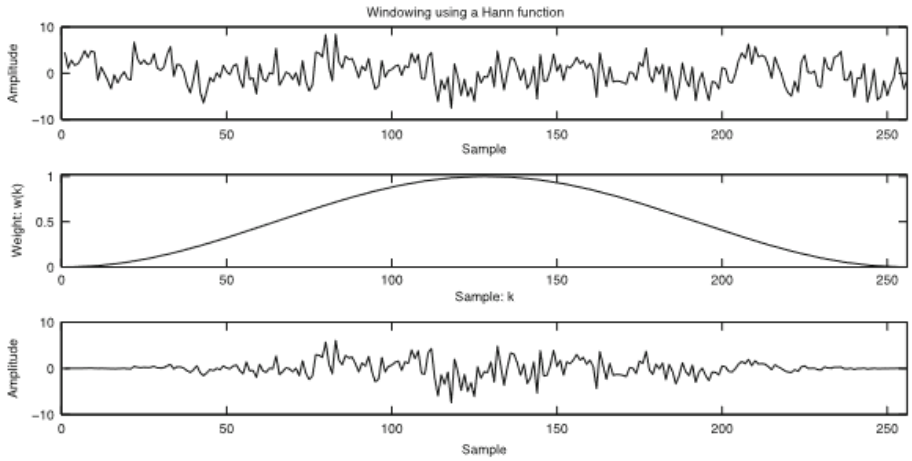


Figure2.4: Windowing a 256 sample frame using the Hann function (Knees & Schedl, 2016)

After applying the window function and computing the fast fourier transform (FFT) of the time domain signal we convert the signal into the frequency domain, after which it becomes easy to calculate audio features. As each of these features is computed over single frames, they are finally combined in order to achieve the feature that represents the entirety of the audio segment or music segment under consideration.

2.3 Machine learning and music

The field of machine learning has developed a lot in recent times and has seen its application in some of the most widely used and powerful technologies of the 21st century, from voice recognition, shopping, music recommendation, self-driving cars, identifying diseases, etc. Machine learning is nothing but the capacity of a computational system to learn patterns and structures from large amounts of the dataset to make predictions and identify new data points. Because of its ability to learn and

execute patterns, machine learning has sparked a lot of interest in the world of music, where musicians and academics have used algorithms to create unique styles or have broken the rules to test the system's capabilities in unanticipated ways. Many a time musicians borrow the existing algorithm to use for pattern recognition tasks, whereas other times they use it in unexpected ways for example in music-making, sound design, or to establish new relationships between machines and musicians. There have been many applications of machine learning in music, (Fiebrink & Caramiaux, 2016) discusses machine learning as a creative music tool for music-making and explore it within the context of human-computer interaction. David cope's early work (Cope, 1996) in algorithmic learning and recreation of western classical music was a pivotal study in music-making using a machine learning algorithm. Rendering and creation of musical scores were explored by (Hiraga et al., 2004), and more machine learning approaches that are useful for music creation are investigated in the case study (*Machine Learning Research That Matters for Music Creation*, n.d.). In the recent times with the advent of deep learning techniques, music-making using the same has also been explored a lot (Sturm et al., 2016) in which the authors have applied a deep learning model called long short term memory networks(LSTM) for music composition, more of such music creation applications have been explored in the survey (Briot et al., 2019). Moreover, machine learning is also widely popular in music information retrieval for signal analysis, music recommendation, and mood and genre classification. Since the early days of the internet it has been studied a lot, (Tzanetakis & Cook, 2002) used rhythmic and harmonic contents and used supervised machine learning approaches such as Gaussian mixture models and k nearest neighbor classifier. Hidden Markov models which are extensively used for speech recognition are also used for classification purposes and are explored by (Scaringella et al., 2006; Shao et al., 2004). Support vector machines with different distance matrices are studied and compared (Mandel & Ellis, 2005)). With the developments in deep learning techniques, researchers have employed deep learning networks as well for the purpose of classification. (Li, Li, et al., 2018; Quinto et al., 2017; Raval, 2021). Most of these techniques follow a fairly standard pipeline with minor changes according to the technique applied. First, the pre-processing of the data in case its complex in nature is required. The complexity could be full quality audio files that are long in duration, or if the metadata or MIDI data is used then manipulation of these input points to make the next step of feature extraction optimum. After acquiring the reduced input, the process of feature extraction is implemented. These features sometimes also have high dimensionality or are not standardized,

so to reduce these anomalies dimensionality reduction techniques and standardization techniques are applied. After getting the simplified feature set, they are fed into the machine learning algorithm for training and learning processes. Now it depends on the type of algorithm used if they need to be given a testing set of data or not. It depends on if the algorithm is supervised, unsupervised or semi-supervised. After this, the predictions or the results are acquired. The optimal balance to strike between machine learning and expert human knowledge is a crucial topic for MIR and machine learning researchers at each phase of the pipeline. In general, the balance for feature extraction has shifted toward expert knowledge, whereas the balance for inference has shifted toward machine learning. For example Researchers working on automatic chord estimation from audio, may know ahead of time that a useful feature to extract is the amount of sound energy in each pitch class (C, C sharp, D, etc.), but when inferring actual chord labels from these so-called chroma vector features, they may prefer to let a machine decide where the thresholds between particular chords should be. (Burgoyne et al., 2015). Finally, it should be noted that the models applied in this thesis are Expectation-Maximization with Gaussian Mixture Models(EM-GMM) for the Barreira's approach, and Self Organizing Maps(SOM) and Growing Heirarchical Self Organizing Maps(GHSOM). Detailed descriptiom of both the algorithms are provided in the later sections.

2.4 Summary

This section established the background information required to understand the systems employed in this thesis project. The section dwelled on important topics of Music Information retrieval, Audio signal processing, and machine learning and its application in music. These are the important topics that are heavily applied in this thesis project. The reader should by now be familiar with the essential concepts required to understand the following sections of the thesis, particularly what is MIR, how are audio signals analyzed and processed, and how machine learning is helping in various MIR topics including audio and music classification which is the main focus of this thesis.

3 Methods

The research question of this thesis is to study unsupervised machine learning techniques for the classification of musical genres, particularly sub-genres of electronic music that are similar to each other in terms of their musical elements. Finding answers to this question requires the study of unsupervised machine learning techniques as well as supervised machine learning techniques to be able to distinguish the process from one other. Along with this, a deep dive into audio signal processing is also required. As mentioned earlier there have been multiple approaches to the problem of music genre classification and new techniques keep getting developed. So, it is important to choose algorithms that are worth exploring and can give results that can be further implemented into real-world applications. A large part of the effort in this thesis has gone into the exploration of these techniques to find the optimum one and then implement them in the context of the research question. Additionally, another effort-consuming task has been to investigate the audio signal processing techniques and learn what audio features to extract and how to extract them. Since the implementation of these concepts is integral to the research process, the research method has been evaluated by emulating approaches widely popular in academia and commercially.

Even though a significant portion of the workload behind this thesis has been the implementation of algorithms and software, it is not a software engineering project. The software emulated is not being studied in and of itself; instead, it is being utilized to generate data that may be used to reflect on the research topic. As a result, typical software engineering assessment metrics like performance efficiency, code quality, and stability aren't as useful in determining the thesis's success. The approaches were implemented to study and explore how well the algorithms perform and if they are viable for real-world applications. It would likely need to be implemented slightly differently if it were to be used in real-world scenarios. A possible continuation of this thesis could be to implement the approaches presented in the classification of genres for an audio streaming app and recommendation system. This continuation would resemble a typical software engineering project in which the topic of research could be the development of efficient tools for users to have easy access to a large archive of heterogeneous music pieces. Such a project would likely benefit

from a redesign, focusing more on the user experience and software efficiency to use it in a real-world framework.

3.1 Choosing the systems

The approaches executed in this thesis are suggested by (Barreira et al., 2011) and (Rauber et al., 2002). In which the first approach uses a model-based approach based on a technique called Expectation-Maximization for Gaussian Mixture Model (EM-GMM), while later uses a technique based on Self-organizing maps (SOMs) and Grow ing Hierarchical Self-Organizing Maps (GHSOMs). Both approaches use low-level audio features to train the models. The first approach was chosen because this algorithm can be used for complex models with a lot of latent variables included. In this model, each cluster formed can have unconstrained covariance. So, the data points do not have a hard assignment to any particular cluster but have a probabilistic assignment, meaning the algorithm provides us with a probability of the data point belonging to a certain cluster and not just places a data point into a cluster that is also called as a hard assignment. This is useful for this research thesis, as the data points dealt with are sub-genre of electronic music, and there can be multiple similarities between them and also have a subjective classification bias. Therefore, a probabilistic assignment helps in the way that if the data point is not assigned to one particular but is allocated to multiple clusters. And, if the user feels the allocation is not appropriate due to subjective bias, the model can be easily redesigned to perform according to the needs of the user. The second approach uses psychoacoustic models and self-organizing maps (SOM) for classification purposes. Being a decidedly stable and flexible model, the SOM has been employed in a wide range of applications, ranging from financial data analysis, via medical data analysis, to time series prediction, industrial control, and many more. It is generally used in applications that have a very high number of distinct data points. It basically offers itself to the organization and interactive exploration of high-dimensional data spaces. Musical genres can be highly high dimensional and with new music added every day it can be used very efficiently for classification and retrieval purposes. However, due to its topological characteristics, the SOM, on the other hand, can also be utilized as an index structure in high-dimensional databases, allowing for scalable proximity searches. As a result, the SOM combines and makes available in a

convenient manner multiple use cases like classification, interactive exploration, and indexing and retrieval of information represented in the form of high-dimensional feature spaces, where exact matches are either impossible due to the fuzzy nature of data representation or the respective type of query, or at the very least computationally prohibitive, making them particularly suitable for musical databases.

3.2 Database selection

MTG-Jamendo is the name of the dataset utilized in this study (Bogdanov et al., 2019). This dataset was chosen as it features high-quality audio for 55,701 entire songs, each lasting at least 30 seconds, and is in MP3 format with a quality of 320kbps and a sample rate of 44.1kHz. This dataset was created using music from the Jamendo platform, which is freely available under creative commons licenses (website). The tracks in the collection have been tagged with 692 tags that include genres, instrumentation, moods, and topics. All tags were provided by the artists that contributed music to Jamendo; however, the dataset's creators pre-processed them for tag purification. This dataset has an archive of over 250 distinct musical genres including a number of sub-genres of music, along with multiple instrument tags and mood or theme-based tags, making it advantageous for the particulars of this project.

3.3 Evaluation techniques

The clustering or classification issue, from an intuitive standpoint, has a very obvious goal: accurately grouping a set of unlabeled data. The concept of "cluster" cannot be fully defined, despite its intuitive appeal, which explains the vast range of clustering methods that have been developed. Jon Kleinberg (Kleinberg, 2002) suggests three axioms that emphasize the features that a grouping issue should have to be regarded as "excellent," regardless of the technique employed to solve it. Scale invariance, consistency, and wealth are the three axioms. Scale invariance simply means when all distances between points are scaled by the factor defined by a constant, this axiom states that a clustering technique should not change its results. When the distances inside clusters decrease and/or the

distances between clusters rise, a clustering process is "consistent" when the clustering results do not alter. Richness or wealth implies that the clustering function must be able to create any arbitrary partitioning or clustering of the given data set. Given the above three axioms, Kleinberg proves the following theorem: For every $n >= 2$, no clustering function f satisfies scale invariance, richness, and consistency altogether. Because the three axioms cannot be satisfied at the same time, clustering algorithms can be devised to violate one while satisfying the other two. When the number of clusters is known ahead of time, clustering algorithms may typically meet the characteristics of scale invariance and consistency by relaxing their richness. Certain algorithms may be tweaked to meet two of the three axioms while ignoring the third (e.g. simple linkage with different stopping criteria) (Palacio-Niño & Berzal, 2019). When examining clustering results, various factors must be considered to ensure that the algorithm's conclusions are legitimate (Kleinberg, 2002; Palacio-Niño & Berzal, 2019). They are but are not limited to

- Determining if the data has a clustering tendency (i.e., whether a non-random structure exists).
- Choosing the right amount of clusters
- Evaluating the quality of the clustering findings in the absence of external data.
- Using external data to compare the outcomes obtained.
- Choosing between two sets of clusters to see which is superior.

Clustering is considered to be good when it has a high separation between clusters and high cohesion within clusters (Handl et al., 2005), rather than dealing with distinct metrics for cohesion and separation, several metrics attempt to quantify separation and cohesion in a single measure. Based on these factors various evaluation metrics have been employed for evaluating the two approaches used in this research project are detailed below as described by (Palacio-Niño & Berzal, 2019).

Silhouette coefficient: The silhouette coefficient is the most frequent method for combining cohesion and separation measures into a single statistic. It is given as :

$$S(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}}$$

where $a(i)$ is the average distance of i from all other points in its cluster and $b(i)$ is the lowest average distance of i from all other points in its cluster. For example, if there are only three clusters A, B, and C, and i belongs to cluster C, $b(i)$ is computed by measuring i's average distance from each point in cluster A, then calculating i's average distance from each and every point in cluster B, and picking the lowest value. The global silhouette coefficient is just the average of the particular silhouette coefficients for each example.

It is defined in the interval $[-1, 1]$ for each example in the data set. $S(i)$ being close to zero means that the points are uniformly distributed throughout the Euclidian space, negative values indicate that the clusters are mixed with each other and are overlapped and positive values indicate high separation between the clusters.

Calisnki-Harabasz coefficient: The variance ratio criteria, often known as CH, is a metric based on the premise that clusters that are compact and well-spaced from one another are desirable clusters. The variance of the sums of squares of individual object distances to their cluster center is divided by the sum of squares of the distance between the cluster centers to determine the index. The better the clustering model, the higher the Calinski-Harabasz Index score. It is defined as

$$CH_k = \frac{BCSM}{k-1} \times \frac{n-k}{WCSM}$$

Here, k denotes the number of clusters and n denotes the number of records in the dataset. The $BCSM$ (between cluster scatter matrix) determines cluster separation, whereas the $WCSM$ (within-cluster scatter matrix) estimates cluster compactness.

There are many more evaluation metrics that can be used to determine the efficiency of the clustering techniques like the Davies-Bouldin Index, F1 score, Accuracy score or confusion matrix, etc. But for the purpose of this research, only the above two explained metrics are used.

4 System description and Implementation

This chapter presents a description of the two approaches to the problem of unsupervised music genre classification applied to the sub-genres of electronic music. They are approaches suggested by (Barreira et al., 2011) and (Rauber et al., 2002). In this chapter, firstly the process of data acquiring, and data pre-processing is discussed, secondly, the features that are to be extracted are presented for both the approaches followed by the extraction pipeline. Finally, the implementation and in-depth description of both systems, including the algorithms applied in each of them are discussed.

4.1 Data acquiring and pre-processing

The dataset used in this project is called MTG-Jamendo dataset (Bogdanov et al., 2019). It contains audio for 55,701 full songs, with a duration of at least 30 seconds, is in the MP3 format, and has the quality of 320kbps at the sampling rate of 44.1kHz. This dataset is built using music publically available under creative commons licenses on the platform called Jamendo(website). The tracks in the dataset are annotated with 692 tags encompassing genres, instrumentation, moods, and themes that have been applied to the music in the dataset. All tags were given by the artists that submitted music to Jamendo, however, they were preprocessed by the dataset's producers with the purpose of tag cleansing. This dataset has been released recently and is a great addition for training and developing various models for all kinds of MIR research. This dataset was chosen as it has a very wide range of genres available, also it has 16,480 tracks in the category of Electronic genre alone with several sub-genres. Therefore a pre-processing of the dataset is essential to better suit the need of this project. The sub-genres chosen for this project are: “house”, “trance”, “funk” and “minimal”. Now since there are thousands of tracks present in the electronic genre category and a few hundred in each subcategory, it

becomes crucial to extract a limited amount of tracks for the testing and training of the algorithm, therefore, to save computational expenses, 100 tracks from each genre are selected randomly. Furthermore, to make it more optimum all the tracks are down sampled to 22,050 Hz and are turned into mono tracks. Finally, the full-length tracks are sliced into a duration of 30 seconds clip is taken for extracting the features and standardization of duration, considering not every track has an equal duration. After the preprocessing is completed, the next step is to extract features from these audio samples.

4.2 Feature extraction

As mentioned earlier, features are the descriptors of the sound extracted from the audio sample, that acquire less storage space and dimensions, but still provide an accurate representation of the characteristics of the audio. In this section, features proposed by Barreira and Rauber for classification purposes are delineated, and some similar features may be seen between the two. The fundamentals discussed previously in the feature extraction techniques are applied here. The python libraries librosa, SciPy, and NumPy are used for the extraction. Features proposed by Barreira and Rauber are discussed in the following sections. The features proposed by Barreira are classified into two categories, the first one is called computational features, meaning they do not represent any musical information but only describe the mathematical analysis of the signal and give important information about the characteristics of the sound. The second category is called perceptual features, these features mathematically represent musical properties based on the human hearing system. To limit the context of this project, only computational features are considered. Computational features are widely employed and have been used in a number of research on automated music genre classification (Cataltepe et al., 2007; Koerich & Poitevin, 2005; Lidy & Rauber, n.d.; McKinney & Breebaart, 2003; Tzanetakis & Cook, 2002) and more. They used the features proposed by (Tzanetakis & Cook, 2002) which are, spectral centroid, spectral roll-off, spectral flux, Mel frequency cepstral coefficients, root mean square energy, and spectral bandwidth and are described below. Moreover, the spectral properties are calculated over each window of the spectrogram and a mean is taken of the calculated values to get one value for the entirety of the audio sample. The features suggested by Rauber are Specific loudness sensation also called Sone that shows the relationship between phon and sone values and Rhythm patterns per frequency band. To simplify the calculations and focus more on the model, another loudness descriptor

called RMS energy values is used instead of Sone values, along with this it is also paired with other widely used features like Mel frequency cepstral coefficients, zero-crossing rate, and power spectral value.

4.2.1 Spectral Centroid (SC)

The spectral centroid represents the center of gravity of the magnitude spectrum. In other words, the frequency band where most of the energy is concentrated (Knees & Schedl, 2016). This property is used to determine the "brightness" of a sound, and so is related to music timbre. The value of spectral centroid is the average frequency weighted by amplitudes, divided by the sum of the amplitudes, and is the individual centroid of a spectral frame, or

$$spectral\ centroid = \frac{\sum_{k=1}^{N} kF[k]}{\sum_{k=1}^{N} F[k]}$$

Here, $F\ [k]$ is the amplitude corresponding to bin k in the discrete fourier transform (DFT) spectrum.

4.2.2 Spectral roll-off

It is defined as the frequency at which a certain proportion (cutoff), generally taken at 85%, of the spectrum's total energy, is stored. The roll-off frequency can be used to distinguish between harmonic (below roll-off) and noisy sounds (above roll-off) (Peeters, 2004).

$$\sum_{0}^{fc} a^2\ (f) = 0.85 \sum_{0}^{sr/2} a^2\ (f)$$

Here fc is the spectral roll-off frequency and $sr/2$ is the Nyquist frequency.

4.2.3 Spectral flux:

The spectral flux is a measure that indicates how quickly a signal's spectral content changes over time. The squared difference between the normalized magnitudes of the spectra of two subsequent short-term windows is used to calculate spectral flux, which measures the spectral change between two frames. It is therefore a frequency domain feature and is computed using the following equation, where D_t is the frame by frame normalized frequency distribution in frame t (Knees & Schedl, 2016).

$$SF_t = \sum_{n=1}^{N} \left(D_t(n) - D_{t-1}(n)\right)^2$$

4.2.4 Zero-crossing rate (ZCR):

The zero-crossing rate is one of the simplest and easiest features to calculate. It measures the number of times the amplitude value changes its sign within the frame t under consideration (Knees & Schedl, 2016). In other terms, it's the number of instances the signal shifts from positive to negative and vice versa, divided by the frame duration. The ZCR is calculated using the following formula:

$$ZCR_t = \frac{1}{2} \times \sum_{k=t\times k}^{(t+1)\times K-1} \left|sgn\big(s(k)\big) - sgn(s(k+1))\right|$$

Where $sgn(.)$ is the sign function, i.e.

$$sgn[x_i(n)] = \begin{cases} 1, & x_i \geq 0 \\ -1, & x_i < 0 \end{cases}$$

The ZCR can be understood as a metric for signal noise. In the event of noisy signals, for example, it frequently has greater values. It also has uses in speech recognition and the detection of percussive sounds, as speech signal typically has lower ZCR value and percussive sounds have higher ZCR values.

4.2.5 Mel frequency cepstral coefficients (MFCC)

The MFCC of a signal is a set of features that describes the overall shape of the spectrum. Successfully used by Davis and Mermelstein in speech recognition tasks for the first time in 1980 (Davis & Mermelstein, 1980), the MFCC coefficients characterize the cepstrum energies on a non-linear scale called the Mel-scale. Mel frequency scale can well reflect the non-linear characteristics of the human auditory system. The cepstrum is the logarithm of the spectrum's Fourier Transform (or Discrete Cosine Transform DCT). The Mel-cepstrum is a cepstrum calculated using Mel-bands rather than the Fourier spectrum, and the MFCC are the coefficients of the Mel cepstrum. This feature is very useful in speech recognition and is also used for determining the timbre of the sound. To calculate MFCC, the first step is to compute the DCT of the frame. Secondly, Mel-spaced filter banks are calculated, these are a group of triangle bandpass filters that simulate the characteristics of the human ear and are applied to the

spectrum of the speech signal. You then take the logarithm of this band and finally the DCT is taken to obtain the MFCC coefficient. The coefficients can be expressed as:

$$C_j = \sum_{j=1}^{K} X_j \cos\left(j(i-1)/2\frac{\pi}{K}\right)$$

Here C_j is the MFCC coefficient, X_j is the power spectrum of Mel frequency, j = 1, 2, 3, …, K (K is the number of desired coefficients).

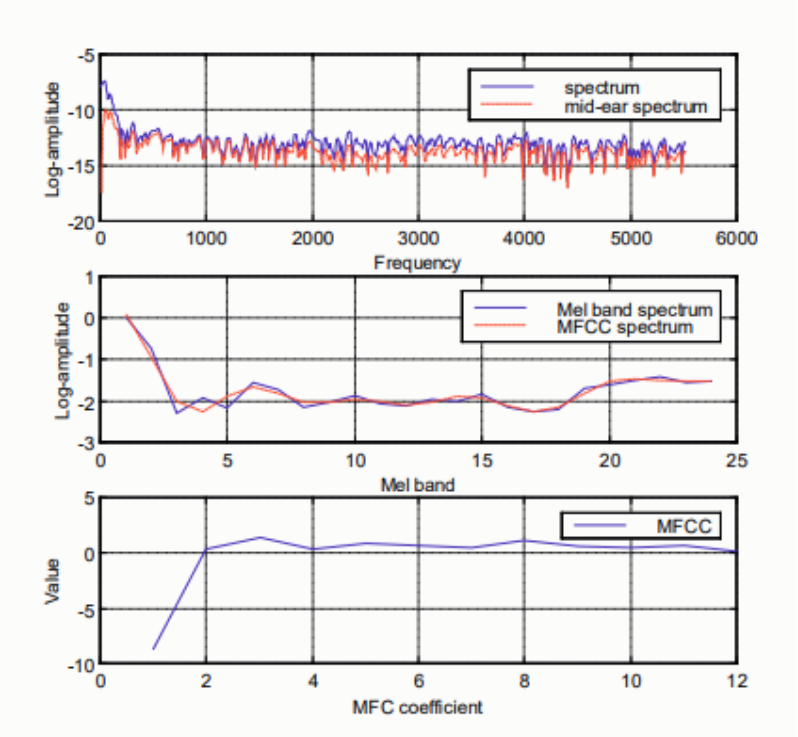


Figure 4.1: [Top]signal spectrum and mid-ear filtered spectrum(dashed line)[middle]Mel band spectrum and MFCC spectrum(dotted line) [bottom]MFCC coefficients (Peeters, 2004)

4.2.6 Root mean square (RMS) energy

Also called RMS level or RMS power, it is a time-domain feature and is used to describe the average amplitude value of the signal. It is used as loudness estimation and as an indicator for new events in the audio sample It is given by:

$$RMS_t = \sqrt{\frac{1}{K} \times \sum_{k=t\times K}^{(t+1)\times(K-1)} s(k)^2}$$

The amplitude of an audio transmission can be both positive and negative. The negative numbers would counterbalance the positive ones and the result would be zero if we calculated the arithmetic mean of a sine wave. This method provides no information about the average signal strength.

Here's when the RMS level comes in handy. It uses a signal's magnitude as a metric for signal intensity, irrespective of if the amplitude is positive or negative. The magnitude is obtained by first squaring each sample value (to make them all positive), then calculating the signal average, and ultimately the square root operation.

4.2.7 Bandwidth

Bandwidth (BW), sometimes also called Spectral Spread (SS), is a frequency domain characteristic that is generated from the spectral centroid. The spectrum range of the signal's important components, i.e. the regions surrounding the centroid, is indicated by spectral bandwidth. It can be regarded as a deviation from the signal's mean frequency (Knees & Schedl, 2016). The average bandwidth of a music piece may serve to describe its perceived timbre. (*An Introduction to Audio Content Analysis*, 2012). It is given as:

$$BW_t = \frac{\sum_{n=1}^{N} |n - SC_t| \cdot m_t(n)}{\sum_{n=1}^{N} m_t(n)}$$

4.2.8 Power spectral density

The power spectral density (PSD) or power density (PD) or power density spectrum is the frequency domain distribution of the average power of a signal $x(t)$. The PSD function is indicated by $S(\omega)$. We may discover the range of power across which the signal frequencies operate by looking at the PSD, which indicates the power of various frequencies contained in the signal. The PSD profile is just a plot of power against frequency.

$$S(\omega) = \lim_{\tau \to \infty} |X(\omega)|^2$$

4.3 Barreira system description

An overview of the system proposed by Barreira is shown in figure 4.2. After extracting the above-mentioned features suggested in the (Barreira et al., 2011) approach to clustering of music, a matrix of the features concerning the audio samples is created followed by standardization and feature reduction techniques. Finally, the clustering is obtained by the Expectation-Maximization with the Gaussian mixture model algorithm.

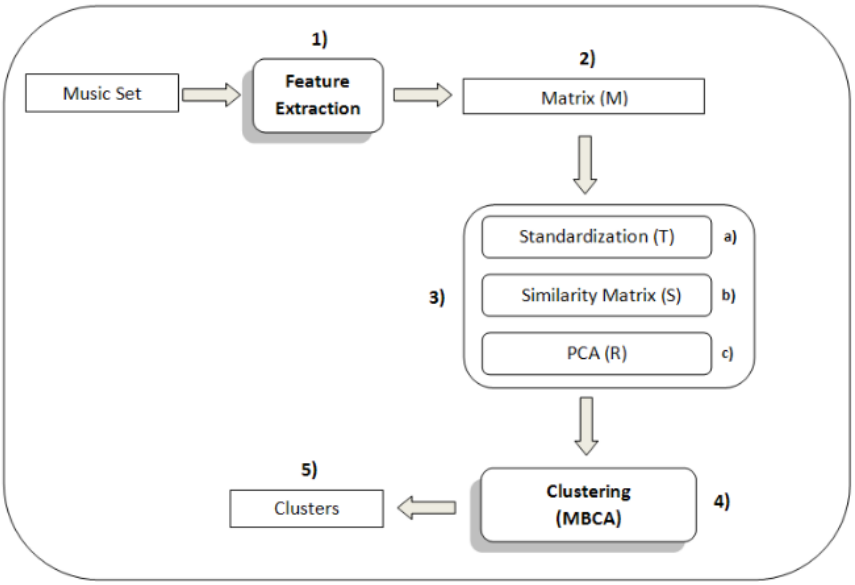


Figure 4.2: System pipeline (Barreira et al., 2011)

4.3.1 Feature matrix and Standardization

First, we start by creating a matrix (M) of the features whose columns correspond to characteristics and whose lines belong to music samples in the training set. Now to set all the features to equal value and scale, standardization of Matrix M is performed, and a new matrix T is created with the same dimension as matrix M, that is, both matrices are $(N \times F)$, where N is the number of samples in the training set and F is the number of features. Now the standardized value of a feature present in the cell $t_{s,f}$ of this new matrix T is given by,

$$t_{s,f} = \frac{m_{s.f} - m_{.,f}}{\sqrt{var(M_f)}}$$

Here, $m_{.,f}$ is the mean value of the fth column of matrix M, that is,

$$m_{.,f} = \frac{1}{N}\sum_{i=1}^{N} m_{i,f}$$

and the variance of feature $var(M_f)$ is obtained from

$$var(M_f) = \frac{1}{N-1}\sum_{i=1}^{N} (m_{i,f} - m_{.,f})^2$$

4.3.2 Dimensionality reduction using Principal Component Analysis (PCA)

After this standardization process, a dimensionality reduction technique called principal component analysis (PCA) is implemented. Its implemented to further reduce the number of features and obtain the ones that provide the best description of the samples and are most helpful for the process of classification. The earliest research published on PCA dates back to the early 1900s (Hotelling, 1933; Pearson, 1901). Its concept is straightforward: lower the dimensionality of a dataset while keeping as much 'variability' (statistical information) as feasible. To achieve this goal, PCA computes new variables called principal components which are generated as linear combinations of the original variables. The first principal component must have the greatest possible variance (i.e., inertia), so this component will 'explain' or 'extract' the majority of the data table's inertia. The second component is computed with the requirements of being orthogonal to the first and having the greatest achievable inertia as explained by (Abdi & Williams, 2010)**.** In the same way, more components are computed.

4.3.3 The Clustering stage

After extracting the features, and performing dimensionality reduction and standardization, we have the inputs prepared for the clustering stage. At the clustering stage, Model-based Clustering Analysis (MBCA) as proposed by (Fraley & Raftery, 1998) is employed. This method makes no assumptions concerning the number of clusters, their structure, or their orientation. It displays the data using a variety of models, each with its own set of geometric features. This method uses a mixture model to describe data, with each element corresponding to a distinct cluster calculated similarly. The cluster formation is done using the EM (expectation-maximization) algorithm for maximum likelihood, based on Gaussian mixture models.

As explained by (Fraley & Raftery, 1998), in model-based clustering, the data are considered to be created by a mix of underlying probability distributions, each of which represents a separate group or cluster. To explain the clustering process, first Gaussian mixture models are delineated followed by the Expectation and Maximization steps to generate clusters.

4.3.3.1 Gaussian Mixture Models

A Gaussian mixture model (GMM) is a probabilistic model in which all data points are created by combining a set amount of Gaussian distributions with uncertain variables. In other words, the data points are not given hard assignments in one mode. Rather they are assigned to multiple models and are given a probabilistic value of their assignment for the corresponding model. As explained by (Guo et al., 2012) a Gaussian Mixture Model is a model that linearly combines various Gaussian distributions. The following is a representation of a GMM with K Gaussian components:

$$p\,(x_n) = \sum_{k=1}^{K} w_k\, G(x_n|\mu_k, \theta_k)$$

Here, $x_n = (x_{n1}, x_{n2}, \dots, x_{nD})$ is an D-attribute vector that represents a data instance. It may be thought of as a point in the Euclidean space of D dimensions. $G(x_n|\mu_k, \theta_k)$ is a Gaussian probability density controlled by a mean vector $\mu_k = (\mu_{k1}, \mu_{k2}, \dots, \mu_{kD})$ and covariance matrix θ_k. Now the probability density function for a GMM can be mathematically defined by

$$G(x_n|\mu_k, \theta_k) = \frac{\exp\{\frac{-1}{2}(x_n - \mu_k)^T\, \theta_k^{-1}(x_n - (x_n - \mu_k))\}}{(2\pi^{\frac{d}{2}} \times |\theta|_k^{1/2})}$$

The probability function $G(x_n|\mu_k, \theta_k)$ is also referred to as the kth Gaussian component of the GMM. Finally, w_k is the mixture coefficient of the kth Gaussian component. The mixture coefficients $w_1, \dots, w_k$ must be non-negative numbers satisfying

$$\sum_{k=1}^{K} w_k = 1$$

To summarize, three-parameter sets drive a Gaussian mixture model: mixture coefficients w_k, mean vectors μ_k and covariance matrix θ_k.

4.3.3.2 Expectation-Maximization for GMMs

The method suggested in the research by (Barreira et al., 2011) for parameter estimation of GMM is called the Expectation-Maximization algorithm, which is a popular way of parameter estimation in machine learning applications. The Expectation-Maximization for Gaussian Mixture Models(EM-GMM) algorithm iteratively evaluates the parameters of a GMM. We start by taking an initial parameter set randomly, then update it by alternating between the two Expectation-Maximization stages. In other words, we first start by randomly choosing some points as the center of the clusters, then for each data point, we calculate the probability of being in each cluster, this is called as Expectation stage. Lastly, using this probability we recalculate the means and variances of the clusters this is the maximization step. We do these iterations until a convergence criterion is met. The Expectation-Maximization steps are mathematically explained below.

Expectation step: Using the current parameter value estimation that is, mixture coefficients w_k , mean vectors μ_k and covariance matrix θ_k , compute a responsibility value r_{nk} for each data instance x_n with regard to each Gaussian component k. So the responsibility value r_{nk} can be defined as

$$r_{nk} = \frac{w_k G(x_n|\mu_k, \theta_k)}{\sum_{j=1}^{K} w_j G(x_n|\mu_j, \theta_j)}$$

Maximization step: We estimate new parameter sets that are updated, mixture coefficients w_k^+, mean vectors μ_k^+ and covariance matrix θ_k^+,

$$w_k^+ = \frac{N_k}{N}$$

$$\mu_k^+ = \frac{\sum_{n=1}^{N} r_{nk} x_n}{N_k}$$

$$\theta_k^+ = \frac{\sum_{n=1}^{N} r_{nk}(x_n - \mu_k^+)(x_n - \mu_k^+)^T}{N_k}, where\ N_k = \sum_{n=1}^{N} r_{nk}$$

We iterate these steps until convergence is obtained. The MBCA approach suggested by (Fraley & Raftery, 1998) after following the creation of all models, MBCA employs the Bayesian Information Criterion (BIC), which gives an estimation of how good is the GMM in terms of clustering the data points. This is done to assess the evidence of clustering for each pair (model, number of clusters), with the bigger the BIC value, the stronger

the evidence for the pair. Following these steps, clusters are automatically formed.

4.4 Rauber system description

The system described by Rauber (Rauber et al., 2002) uses a widely popular neural network algorithm called Self Organizing Maps (SOM) (Kohonen & Somervuo, 1998), along with its extension called Growing Hierarchical - Self Organizing Maps (GHSOM) (Dittenbach et al., 2000). As mentioned in the feature extraction section, The features recommended by Rauber are specific loudness sensation, also known as Sone, which depicts the link between phon and sone values, and rhythm patterns per frequency band. Instead of Sone values, another loudness descriptor termed as RMS energy values is utilized to simplify the computations and focus more on the model, along with Mel frequency cepstral coefficients and the zero-crossing rate which were also used in the Barreira approach, and the power spectral value is used as well. After extracting these features, they are fed into the SOM and GHSOM algorithms. Before we feed the features input into these algorithms, it is essential to understand the fundamentals of SOM and GHSOM, which are delineated in the following sections.

4.4.1 Self Organizing Maps (SOM)

The self-organizing maps were first proposed by Kohonen in the 1980s (Kohonen, 1982) and have been widely applied since then, and are one of the most distinguishable models of unsupervised artificial neural networks. Inspired by the biological models of neural systems from the 1970s, SOM uses an unsupervised learning technique and a competitive learning algorithm to train its network. SOM is used in clustering and mapping (or dimensionality reduction) techniques to map multidimensional data onto lower-dimensional data, making it easier to grasp complicated situations. It essentially does cluster analysis by mapping high-dimensional input data into a typically 2-dimensional output space while retaining as many topological links between the input data items as feasible. To put it another way, the SOM projects the data space onto a two-dimensional map space in such a way that related data items on the map are close to each other.(Dittenbach et al., 2000; Kohonen, 1982; Rauber et al., 2002)

The SOM is, in more formal terms, consists of two layers. The input layer is the first layer and has all the feature points, and the second one is the output layer which is also known as the output lattice. This output layer consists of a set of units i which are ordered according to some topology, with a two-dimensional grid being the most prevalent choice. A model vector m_i is assigned to each of the units i, having the same dimension as the input data. Starting with randomly initializing the weight to vectors, the mapping phases of SOM begin. Following that during each learning step t , an input pattern $x(t)$ is randomly selected from the set of input vectors and presented to the map. Next, the unit showing the most similar model vector with respect to the presented input signal is selected as the winner c, where Euclidean distance is a popular choice for computing similarity.

$$c(t) : \left||x(t) - m_c(t)|\right| = \min_i\{||x(t) - m_i(t)||\}$$

Adaptation of the weights occurs at each learning iteration and is defined as a steady decrease of the difference between the input vector and the model vector's corresponding components. The degree of adaptation is governed by a monotonically declining learning rate, resulting in big adaptation steps at the start of the training session, followed by a fine-tuning phase at the conclusion. Units in a time-varying and the steadily shrinking neighborhood surrounding the winner are also modified. This allows for a spatial organization of the input patterns, with similar inputs being mapped to locations in the grid of output units that are adjacent to one other. As a result of the self-organizing map's training process, the input patterns are ordered topologically. The neighborhood of units around the winner is described using a neighborhood kernel h_{ci}, which takes into account the distance between the unit i under examination and the current learning iteration's winner unit c. A popular representation to define the structure of the neighborhood kernel is the Gaussian model, which ensures the units closest to the winning unit have the most adaptation of weight. It is common at the beginning stages to select the kernel large enough to cover a wide area of the units in the map, which is then gradually shortened in such a way that only the winner unit is adapted in the end and neighbor units remain more or less untouched.

By combining the principles mentioned above, the learning rule can be given as:

$$m_i(t+1) = m_i(t) + \alpha(t) \cdot h_{ci}(t) \cdot [x(t) - m_i(t)]$$

Here, α represents the time-varying learning rate, h_{ci} represents the neighborhood updating with time, x is the input pattern presented in the current iteration and m_i defining the model vector assigned to unit i.

A simple graphical explanation of the architecture of the self-organizing map is given by (Rauber et al., 2002) and is shown in the figure below.

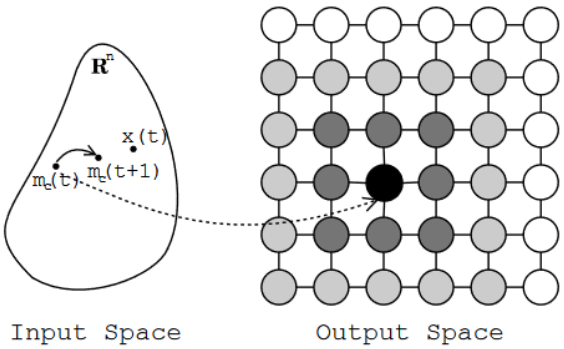


Figure 4.3: A simple depiction of SOM training and model vector adaptation (Rauber et al., 2002)

The output space in this figure is made up of a square of 36 units, shown as circles, producing a grid of 6×6 units. One input vector $x(t)$ is chosen randomly and is mapped on the grid of the output units. The winner c with the greatest activation level is chosen. Consider the winning unit, which is illustrated in the figure as the black unit. The model vector $m_c(t)$ is then moved towards the input vector, as shown in the figure by the arrow in the input space. As a result of this adjustment, the unit c will produce an even higher activation function with respect to the input pattern x at the next learning iteration, $t+1$, because the unit's updated model vector $m_c(t+1)$ is now nearer to the input pattern x in terms of the input space. Aside from the winner, surrounding units are also subjected to alteration. The graphic depicts units that are subject to adaptation as shaded units. The level of adaptation, and hence the spatial breadth of the neighborhood kernel, is shown by the coloring of the different units. In general, units in close proximity to the winner are more firmly adapted, and as a result, they are portrayed in the image with a darker hue. In this way by rearranging the input vectors around the winning processing unit, the clusters are formed.

4.4.2 Growing Hierarchical Self Organizing Maps

As described by (Dittenbach et al., 2000), the Growing Hierarchical Self-Organizing map (GH-SOM) is based on the usage of a hierarchical structure with numerous levels, each of which includes a collection of separate Self-Organizing Maps (SOMs). At the top of the hierarchy, one SOM is utilized. A SOM might be added to the next tier of the hierarchy for each unit in this map. This approach is replicated with the GHSOM's third and subsequent levels. We choose to utilize an incrementally expanding version of the SOM because one of its drawbacks is its fixed network design. This spares us the task of determining the network size ahead of time, as this is done during the unsupervised training phase. We begin with layer 0, which is made up of just one unit. The mean of all input data is used to establish the weight vector of this unit. The training procedure begins with a small map in layer 1 of, say, 2 × 2 units that self-organizes using the usual SOM training method.

To briefly summarize the SOM training method, a randomly chosen input pattern is supplied to the neural network. The distance between each unit's weight vector and the input vector determines its activation. The unit with the shortest distance, i.e. the winner, as well as a number of units near the winner, are modified. The difference between the vector's components is gradually reduced throughout the adaptation. After adaptation, the winner will resemble the input pattern more closely. This training procedure is performed for a predetermined number of iterations defined as λ. After λ training repetitions, the unit with the greatest difference between its weight vector and the input vector represented by this unit is chosen as the error unit. A new row or column of units is introduced between the error unit and its most different neighbor in terms of the input space. These new units' weight vectors are set to the average of their neighbors.

The quantization error q_i is a straightforward criterion for guiding the training procedure. It's determined as the total of the distances between a unit i weight vector and the input vectors mapped onto that unit. It can also be used to assess a SOM's mapping quality based on the mean quantization error (MQE) of all units on the map. The better the map is trained, the lower the QE value. A map expands until its MQE is reduced to a fraction τ_1 of the q_i of the unit i in the hierarchy's previous layer. As a result, the map now depicts the data that was mapped onto the upper layer unit i in more detail.

As previously stated, the GHSOM's basic architecture comprises one SOM. In the case of heterogeneous input data being mapped on a single unit, this design is enlarged by another layer. A relatively large quantization error q_i, which is over a threshold τ_2, identifies these units. As a percentage of the original quantization error at layer 0, this threshold fundamentally specifies the required resolution level of data representation. A new map will be introduced to the hierarchy in this scenario, and the input data mapped on the relevant upper layer unit will be self-organized in this new map. It again rises until its MQE is reduced to a fraction τ_1 of the quantization error q_i of the upper layer unit in question. It should be noted that this does not always result in a balanced hierarchy. The hierarchy's depth will represent the unpredictability that should be expected in real-world data sets. Based on the desired proportion τ_1 of MQE minimization, we may end up with a very deeper hierarchy of tiny maps, a shallow structure with huge maps, or – in the worst-case scenario – simply one enormous map. When there are no more units available for expansion, the hierarchy's growth comes to an end. A graphical representation as presented by (Dittenbach et al., 2000)of the architecture of GHSOM is shown in figure 4.4 below.

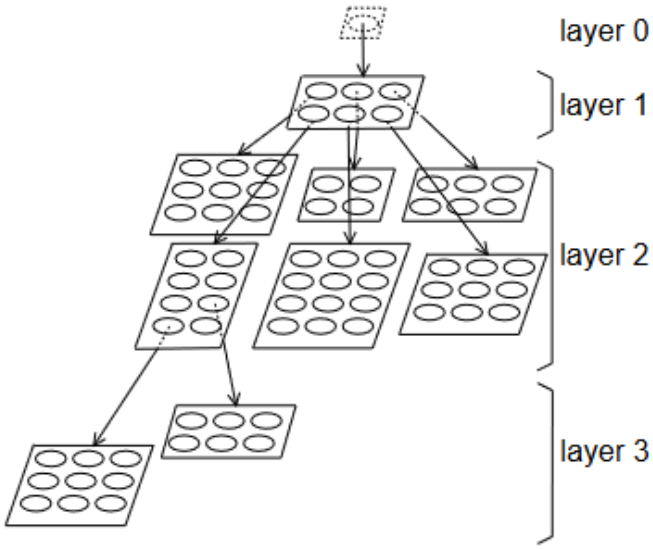


Figure 4.4: Architecture of a GHSOM (Dittenbach et al., 2000)

The figure has one map in layer 1 consisting of 3×2 units and It gives a rough arrangement of the input data's key clusters. The second layer's six separate maps provide a more thorough look at the data. To offer adequate input data representation, two units from one of the second layer maps were further enlarged into third-layer maps. A global orientation of the

newly added maps may be achieved by employing suitable initialization of the maps added at each tier in the hierarchy depending on the parent unit's neighbors. As a result, comparable data will be discovered on surrounding map boundaries in the hierarchy.

4.4.3 The Clustering stage

After extracting the feature vectors that were mentioned above, a dimensionality reduction technique, principal component analysis, that was applied in the approach suggested by (Barreira et al., 2011) is applied here as well. After following these procedures, first, a SOM is trained and results are obtained. Followed by it GHSOM is trained to gain a hierarchical map interface to the music archive by feeding in the reduced feature data obtained by PCA. The GHSOM can be used to create flat maps, similar to traditional SOMs, or to build linear tree structures in addition to hierarchical representations. However, in this project, the experiments are done on the MTG-Jamendo dataset by choosing 4 different sub-genres of electronic music, and results are obtained.

5 Experiments and Results

This section presents the performance evaluation of the two approaches executed in this project. Firstly, the feature selection is reflected upon and the process of their extraction is reflected. Later the results and findings of the two approaches are described. As mentioned in the Methods section the evaluation is done based on two metrics Silhouette coefficient and the Calisnki-Harabasz coefficient. Moreover, 3D figures showing clustering are also presented. In the following sections, the experiments done and the results obtained are outlined for both the approaches

Both of the approaches start by drawing out 100 songs of each genre "house", "trance", "funk" and "minimal" from the MTG Jamendo's dataset. These songs are diced and limited to the first 30 seconds. Moreover, they are downsampled into 22050Khz quality and are turned into mono configuration. Following this procedure, they are sent to the feature extraction pipeline. Once in the pipeline, the computational features mentioned in the system description are evaluated. A graph between the features "RMS" and "MFCC" is plotted to visualize the distribution of the dataset for both the approaches and can be seen in figure 5.1 given below

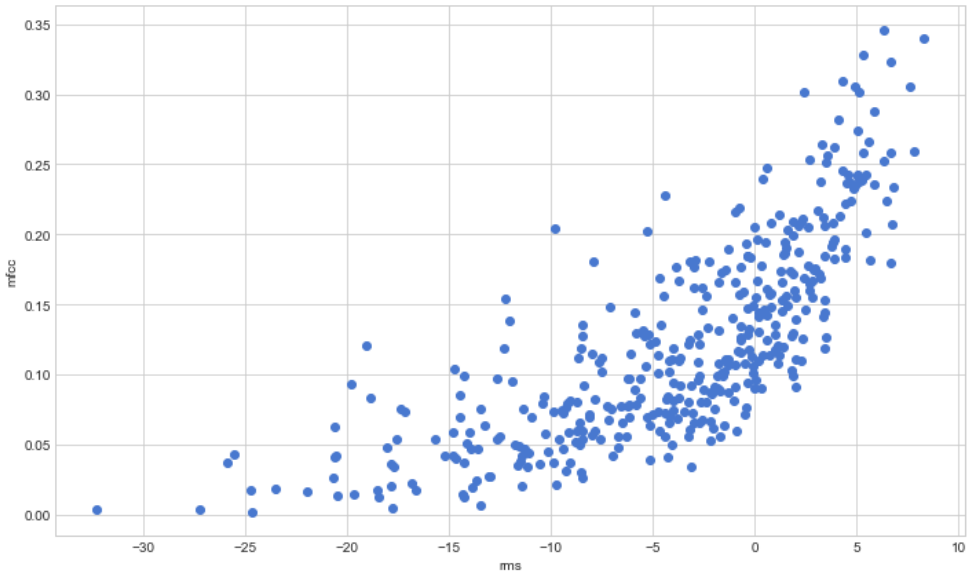


Figure 5.1: Dataset distribution based on audio samples RMS and MFCC values

Additionally, the same distribution is also plotted in 3-Dimension for better visualization. This time the features selected are root mean square energy (RMS), zero-crossing rate (ZCR), and Mel frequency cepstral coefficients (MFCC). These features are chosen because of their accuracy in showing the audio characteristics like amplitude value, speech content, and Timbre of the sound respectively. This can be seen for both the approaches in figure 5.2 given below

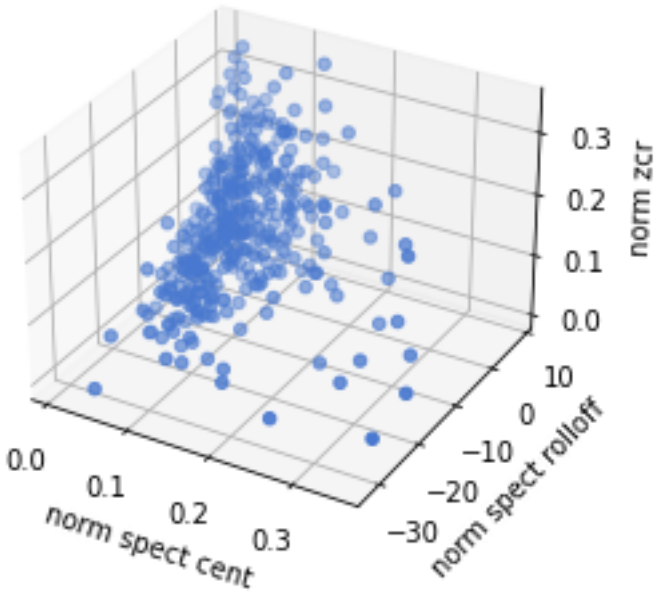


Figure 5.2: Dataset distribution in 3D based on audio sample's RMS, ZCR, and MFCC values

Following this visualization, a standardization of the feature set and a dimensionality reduction technique, principal component analysis (PCA) is employed and a feature set with the most optimum two features is obtained. It should be noted that both the models are fitted and trained on the same data.

5.1 Barreira's model-based approach

After getting the reduced and optimized datasets, the models are built and fitted. Then, the evaluation metrics are obtained. The silhouette coefficient for Barreira's approach is computed to be 0.46. Note that the silhouette coefficient lies between -1 to 1. Where values close to zero indicate that the points are evenly dispersed across Euclidian space, whereas negative

values indicate that the clusters are mixed and overlapped, and positive values show that the clusters are well separated.

Moreover, the Calisnki-Harabasz(CH) coefficient for Barreira's method is calculated to be approximately 883.5. The higher the value of the CH coefficient the better the clustering algorithm is. By observing the values of both the evaluation metrics and matching them to other successful research projects using the same evaluation metrics, it is acknowledged that the approach has above-average performance and is very successful in clustering the sub-genres of electronic music provided. Moreover, a visual classification obtained by projecting the predicted labels on a 3D plane is given in the figure 5.3 below. In the figure, the distinction between the clusters is easily observed.

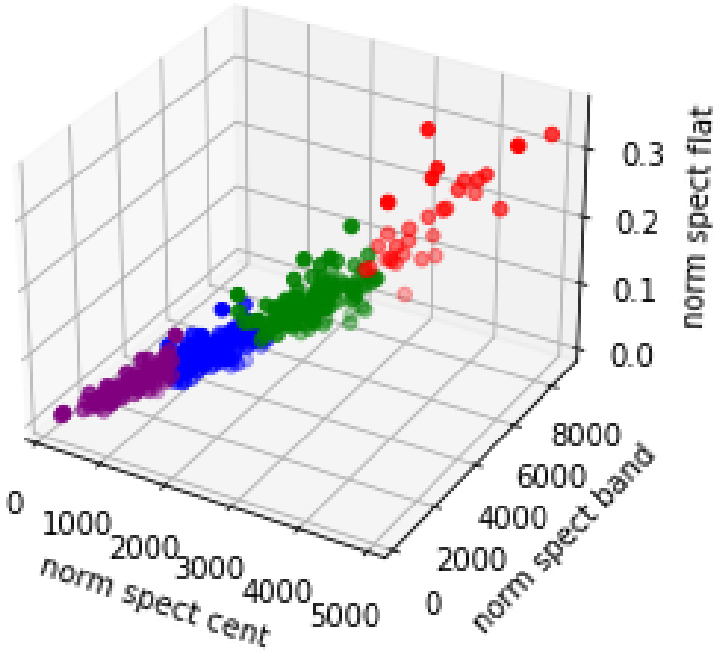


Figure 5.3: Clusters generated by Barreira's clustering approach (EM-GMM)

5.2 Rauber's Self Organizing Maps and the Growing hierarchical Self Organizing Maps

In this approach first Self Organizing Maps are employed, followed by the proposed Growing Hierarchical Self Organizing Maps. Since both the models are fitted on the same data The silhouette coefficient for the Rauber's

approach has a lower value in comparison to the Bareirra's approach and is calculated to be approximately 0.25, while the Calisnki-Harabasz(CH) coefficient is 642.5 approximately.

Moreover, the Growing hierarchical Self Organizing Maps (GHSOM) is evaluated based on the mean square error obtained after fitting the data, and the lower the error value the better the clustering is, and in the case of GHSOM the value is turned out to be 1.728 which is very high from zero error. Therefore it can be observed that with the current hyperparameter values GHSOM has performed poorly in comparison to the easier SOM as well as Barreira's model-based approach. A significant overlap between the clusters is observed. This is also evident in the visualization of the clusters as can be seen in figure 5.4 given below

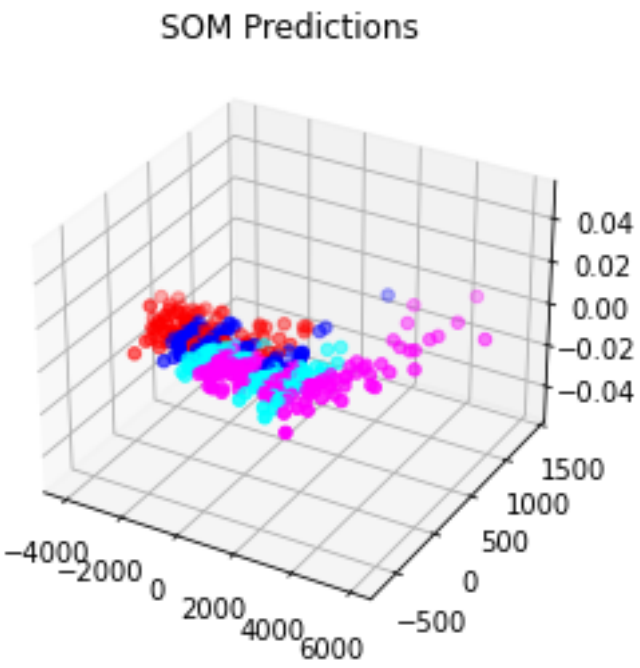


Figure 5.4: Clusters generated by Rauber's clustering approach (SOM)

A comparison between the evaluation metrics of both approaches is given in table 5.1 below. It can be observed that with the current hyperparameter standards Expectation-Maximization with Gaussian Mixture Models is more accurate and efficient in the classification and clustering task in comparison to the Self Organized Maps and Growing Heirarchihcal Self Organizing maps.

	Silhouette coefficient	Calisnki-Harabasz(CH) coefficient
Barreira's Expectation-Maximization with Gaussian Mixture Models(EM-GMM)	0.46	883.5
Rauber's Self Organizing Maps (SOM)	0.25	642.5

Table 5.1: Comparison between evaluation metrics of both the approaches

6 Conclusions

The objective of this research thesis was to explore unsupervised machine learning techniques for music genre clustering and classification and apply them to a much niche context, which is, the clustering of sub-genres of electronic music. These kinds of unsupervised approaches are advantageous because they do not need any information about the labels of data, thereby removing the laborious task of manually tagging the data. Therefore, this method is beneficial as it is totally free from the influences of human bias. Since the tagging of intricate data points such as musical genre can be very subjective, previously labeled data might not provide an accurate representation of the dataset needed to be classified. Furthermore, unsupervised techniques are very advanced in finding new patterns so they can be helpful in finding new genres that are very vague to be classified by humans, which is not achievable with supervised algorithms due to their more static nature. Additionally, it is important to note that there are several approaches that can be used for the problem of genre classification and only a few are explored in this short-term thesis project.

For the categorization of music genres, several clustering and classification approaches have been devised and used. However, the two unsupervised classification techniques applied in this research paper are proposed by Barreira and Rauber. The first one uses a model-based approach that employs an algorithm based on a Gaussian mixture model whose hyperparameters are estimated by the Expectation-Maximization method. The latter approach uses Self Organizing Maps and its extension Growing Hierarchical Self Organizing Maps for the purpose of clustering and classification. The feature set fed into both of these algorithms comprises of low-level audio feature set, also referred to as computational features, which are mathematical representations of the characteristics of the audio sample and are therefore for the most part fortified from human biases. Moreover, these approaches are used because the algorithm proposed by Barreria is based on probabilistic assignment, therefore useful for the vague nature of genre distinction. On the other hand, Rauber’s approach is very

useful in the classification and retrieval of large datasets, thereby useful for the ever-increasing music database and genre categories.

It is observed that both the approaches are successful in clustering the dataset, with one performing better than the other. Barreira's model-based approach has shown better performance with optimum desired values of the evaluation metrics. It is also evident in the visualization of the clusters generated. Moreover, Rauber's approach has shown poor results for the clustering tasks, when employed with the current hyperparameters. The time limitation and scope made it increasingly difficult to employ any hyperparameter techniques for the computation of a better hyperparameter set. This limitation is also discussed in the later section.

6.1 Limitations and Future Work

With limited prior knowledge of machine learning and audio signal processing techniques, the core challenge of the project was to thoroughly understand the algorithms and audio features so that an optimum implementation can be achieved. With only one semester to conduct the research, these difficulties were magnified, resulting in a number of technical and methodological compromises. Although both the techniques were successful to a large extent in the classification of the given four different types of sub-genres of electronic music. It should be noted that the dataset fed into the system was still quite small and the approaches for future work, need to be tested on larger datasets with more data points as well as a greater number of subgenres of electronic music. Additionally, hyperparameter tuning and optimization are highly required to make the models and algorithms robust and perform even better. Moreover, applying them to sub-genres of other popular genres like Classical, Jazz, Rock, etc can also yield interesting results. Redesigning the approaches so that they can be deployed and migrated in a real-life application should be seen as an aspect of future development. Such that they can be used in distinct Music Information Retrieval concepts like a song retrieval system based on genres, or for building a personalized music recommendation system.

References

Abdi, H., & Williams, L. J. (2010). Principal component analysis. *WIREs Computational Statistics*, *2*(4), 433–459. https://doi.org/10.1002/wics.101

Ahmad, A. N., Sekhar, C., & Yashkar, A. (2014). Music Genre Classification Using Music Information Retrieval and Self Organizing Maps. In M. Pant, K. Deep, A. Nagar, & J. C. Bansal (Eds.), *Proceedings of the Third International Conference on Soft Computing for Problem Solving* (pp. 625–634). Springer India. https://doi.org/10.1007/978-81-322-1768-8_55

An Introduction to Audio Content Analysis (1st ed.). (2012). John Wiley & Sons, Ltd. https://doi.org/10.1002/9781118393550

Asim, M., & Ahmed, Z. (2017). Automatic Music Genres Classification using Machine Learning. *International Journal of Advanced Computer Science and Applications*, *8*(8). https://doi.org/10.14569/IJACSA.2017.080844

Aucouturier, J.-J., & Pachet, F. (2003). Representing Musical Genre: A State of the Art. *Journal of New Music Research*, *32*(1), 83–93. https://doi.org/10.1076/jnmr.32.1.83.16801

Bahuleyan, H. (2018). Music Genre Classification using Machine Learning Techniques. *ArXiv:1804.01149 [Cs, Eess]*. http://arxiv.org/abs/1804.01149

Barreira, L., Cavaco, S., & da Silva, J. F. (2011). Unsupervised Music Genre Classification with a Model-Based Approach. In L. Antunes & H. S. Pinto (Eds.), *Progress in Artificial Intelligence* (pp. 268–281). Springer. https://doi.org/10.1007/978-3-642-24769-9_20

Bogdanov, D., Won, M., Tovstogan, P., Porter, A., & Serra, X. (2019). *The MTG-Jamendo dataset for automatic music tagging*.

Briot, J.-P., Hadjeres, G., & Pachet, F.-D. (2019). *Deep Learning Techniques for Music Generation—A Survey* (arXiv:1709.01620). arXiv. https://doi.org/10.48550/arXiv.1709.01620

Burgoyne, J. A., Fujinaga, I., & Downie, J. S. (2015). Music Information Retrieval. In *A New Companion to Digital Humanities* (pp. 213–228). John Wiley & Sons, Ltd. https://doi.org/10.1002/9781118680605.ch15

Cataltepe, Z., Yaslan, Y., & Sonmez, A. (2007). Music Genre Classification Using MIDI and Audio Features. *EURASIP Journal on Advances in Signal Processing*, *2007*(1), 036409. https://doi.org/10.1155/2007/36409

Cope, D. (1996). *Experiments in musical intelligence* (Vol. 12). AR editions.

Costa, Y. M. G., Oliveira, L. S., Koerich, A. L., Gouyon, F., & Martins, J. G. (2012). Music genre classification using LBP textural features. *Signal Processing*, *92*(11), 2723–2737. https://doi.org/10.1016/j.sigpro.2012.04.023

Daniel, F. P., & Cazaly, D. (2000). A Taxonomy of Musical Genres. *In Proc. Content-Based Multimedia Information Access (RIAO*.

Davis, S., & Mermelstein, P. (1980). Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, *28*(4), 357–366. https://doi.org/10.1109/TASSP.1980.1163420

Dittenbach, M., Merkl, D., & Rauber, A. (2000). The growing hierarchical self-organizing map. *Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks. IJCNN 2000. Neural Computing: New Challenges and Perspectives for the New Millennium*, *6*, 15–19 vol.6. https://doi.org/10.1109/IJCNN.2000.859366

Downie, J. S. (2004). The Scientific Evaluation of Music Information Retrieval Systems: Foundations and Future. *Computer Music Journal*, *28*(2), 12–23.

Fiebrink, R., & Caramiaux, B. (2016). *The Machine Learning Algorithm as Creative Musical Tool* (arXiv:1611.00379). arXiv. http://arxiv.org/abs/1611.00379

Fraley, C., & Raftery, A. E. (1998). How Many Clusters? Which Clustering Method? Answers Via Model-Based Cluster Analysis. *The Computer Journal*, *41*(8), 578–588. https://doi.org/10.1093/comjnl/41.8.578

Gambardella, G. (1968). Time Scaling and Short-Time Spectral Analysis. *The Journal of the Acoustical Society of America*, *44*(6), 1745–1747. https://doi.org/10.1121/1.1911332

Gambardella, G. (1971). A contribution to the theory of short-time spectral analysis with nonuniform bandwidth filters. *IEEE Transactions on Circuit Theory*, *18*(4), 455–460. https://doi.org/10.1109/TCT.1971.1083298

Goldman-Eisler, F. (1958). Speech Analysis and Mental Processes. *Language and Speech*, *1*(1), 59–75. https://doi.org/10.1177/002383095800100105

Guo, C., Fu, H., & Luk, W. (2012). A fully-pipelined expectation-maximization engine for Gaussian Mixture Models. *2012 International Conference on Field-Programmable Technology*, 182–189. https://doi.org/10.1109/FPT.2012.6412132

Handl, J., Knowles, J., & Kell, D. B. (2005). Computational cluster validation in post-genomic data analysis. *Bioinformatics*, *21*(15), 3201–3212.

Hiraga, R., Bresin, R., Hirata, K., & Katayose, H. (2004). Rencon 2004: Turing test for musical expression. *Proceedings of the 2004 Conference on New Interfaces for Musical Expression*, 120–123.

Hotelling, H. (1933). Analysis of a complex of statistical variables into principal components. *Journal of Educational Psychology*, *24*(6), 417.

Kageyama, T. (1993). Melody retrieval with humming. *Proc. ICMC 1993*.

Kassler, M. (1966). Toward Musical Information Retrieval. *Perspectives of New Music*, *4*(2), 59–67. https://doi.org/10.2307/832213

Keum, J., & Lee, H. (2006). Speech/Music Discrimination Based on Spectral Peak Analysis and Multi-layer Perceptron. *2006 International Conference on Hybrid Information Technology*, *2*, 56–61. https://doi.org/10.1109/ICHIT.2006.253589

Kleinberg, J. (2002). An impossibility theorem for clustering. *Advances in Neural Information Processing Systems*, *15*.

Knees, P., & Schedl, M. (2016). *Music Similarity and Retrieval* (Vol. 36). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-662-49722-7

Koerich, A. L., & Poitevin, C. (2005). Combination of homogeneous classifiers for musical genre classification. *2005 IEEE International Conference on Systems, Man and Cybernetics*, *1*, 554-559 Vol. 1. https://doi.org/10.1109/ICSMC.2005.1571204

Kohonen, T. (1982). Self-organized formation of topologically correct feature maps. *Biological Cybernetics*, *43*(1), 59–69.

Kohonen, T., & Somervuo, P. (1998). Self-organizing maps of symbol strings. *Neurocomputing*, *21*(1), 19–30. https://doi.org/10.1016/S0925-2312(98)00031-9

Landau, H. J. (1967). Sampling, data transmission, and the Nyquist rate. *Proceedings of the IEEE*, *55*(10), 1701–1706. https://doi.org/10.1109/PROC.1967.5962

Li, Y., Li, X., Zhang, Y., Wang, W., Liu, M., & Feng, X. (2018). Acoustic Scene Classification Using Deep Audio Feature and BLSTM Network. *2018 International Conference on Audio, Language and Image Processing (ICALIP)*, 371–374. https://doi.org/10.1109/ICALIP.2018.8455765

Li, Y., Zhang, X., Jin, H., Li, X., Wang, Q., He, Q., & Huang, Q. (2018). Using multi-stream hierarchical deep neural network to extract deep audio feature for acoustic event detection. *Multimedia Tools and Applications*, *77*(1), 897–916. https://doi.org/10.1007/s11042-016-4332-z

Lidy, T., & Rauber, A. (n.d.). *EVALUATION OF FEATURE EXTRACTORS AND PSYCHO-ACOUSTIC TRANSFORMATIONS FOR MUSIC GENRE CLASSIFICATION*. 8.

Logemann, G. W. (1967). The Canons in the Musical Offering of JS Bach: An Example of Computational Musicology. *Elektronische Datenverarbeitung in Der Musikwissenschaft*, 63–87.

Machine learning research that matters for music creation: A case study. (n.d.). Retrieved May 24, 2022, from http://www.tandfonline.com/doi/epub/10.1080/09298215.2018.1515233?needAccess=true&

Madden, J. P., & Feth, L. L. (2018, December 21). *Temporal Resolution in Normal-Hearing and Hearing-Impaired Listeners Using Frequency-Modulated Stimuli* (world) [Research-article]. ASHA Wire; American

Speech-Language-Hearing Association. https://doi.org/10.1044/jshr.3502.436

Mandel, M. I., & Ellis, D. P. (2005). *Song-level features and support vector machines for music classification*.

McKinney, M., & Breebaart, J. (2003). *Features for audio and music classification*. https://jscholarship.library.jhu.edu/handle/1774.2/22

Miller, G. A. (1951). The Perception of Speech. In *Language and communication* (pp. 47–79). McGraw-Hill. https://doi.org/10.1037/11135-003

Moore, J. K., & Linthicum, F. H. (2007). The human auditory system: A timeline of development. *International Journal of Audiology*, *46*(9), 460–478. https://doi.org/10.1080/14992020701383019

Moorer, J. A. (1975). *On the segmentation and analysis of continuous musical sound by digital computer.* Stanford University.

Palacio-Niño, J.-O., & Berzal, F. (2019). *Evaluation Metrics for Unsupervised Learning Algorithms* (arXiv:1905.05667). arXiv. http://arxiv.org/abs/1905.05667

Pearson, K. (1901). LIII. On lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, *2*(11), 559–572. https://doi.org/10.1080/14786440109462720

Peeters, G. (2004). A large set of audio features for sound description (similarity and classification) in the CUIDADO project. *CUIDADO Ist Project Report*, *54*(0), 1–25.

Pelchat, N., & Gelowitz, C. M. (2020). Neural Network Music Genre Classification. *Canadian Journal of Electrical and Computer Engineering*, *43*(3), 170–173. https://doi.org/10.1109/CJECE.2020.2970144

Pérez-Sancho, C., Rizo, D., & Iñesta, J. M. (2009). Genre classification using chords and stochastic language models. *Connection Science*, *21*(2–3), 145–159. https://doi.org/10.1080/09540090902733780

Poria, S., Gelbukh, A., Hussain, A., Bandyopadhyay, S., & Howard, N. (2013). Music Genre Classification: A Semi-supervised Approach. In J. A. Carrasco-Ochoa, J. F. Martínez-Trinidad, J. S. Rodríguez, & G. S. di Baja (Eds.), *Pattern Recognition* (pp. 254–263). Springer. https://doi.org/10.1007/978-3-642-38989-4_26

Quinto, R. J. M., Atienza, R. O., & Tiglao, N. M. C. (2017). Jazz music sub-genre classification using deep learning. *TENCON 2017 - 2017 IEEE Region 10 Conference*, 3111–3116. https://doi.org/10.1109/TENCON.2017.8228396

Rao, P. (2008). Audio signal processing. In *Speech, Audio, Image and Biomedical Signal Processing using Neural Networks* (pp. 169–189). Springer.

Rauber, A., Pampalk, E., & Merkl, D. (2002). *Using Psycho-Acoustic Models and Self-Organizing Maps to Create a Hierarchical Structuring of Music by Musical Styles.*

Raval, M. (2021). MUSIC GENRE CLASSIFICATION USING NEURAL NETWORKS. *International Journal of Advanced Research in*

Computer Science, *12*(5), 12–18. https://doi.org/10.26483/ijarcs.v12i5.6771

Rihaczek, A. (1968). Signal energy distribution in time and frequency. *IEEE Transactions on Information Theory*, *14*(3), 369–374.

Scaringella, N., Zoia, G., & Mlynek, D. (2006). Automatic genre classification of music content: A survey. *IEEE Signal Processing Magazine*, *23*(2), 133–141. https://doi.org/10.1109/MSP.2006.1598089

Schedl, M., Gómez, E., & Urbano, J. (2014). *Music information retrieval: Recent developments and applications*. Now Publ.

Shao, X., Xu, C., & Kankanhalli, M. S. (2004). Unsupervised classification of music genre using hidden Markov model. *2004 IEEE International Conference on Multimedia and Expo (ICME) (IEEE Cat. No.04TH8763)*, *3*, 2023-2026 Vol.3. https://doi.org/10.1109/ICME.2004.1394661

Sharma, G., Umapathy, K., & Krishnan, S. (2020). Trends in audio signal feature extraction methods. *Applied Acoustics*, *158*, 107020. https://doi.org/10.1016/j.apacoust.2019.107020

Slawson, A. W. (1968). Vowel quality and musical timbre as functions of spectrum envelope and fundamental frequency. *The Journal of the Acoustical Society of America*, *43*(1), 87–101.

Song, Y., & Zhang, C. (2008). Content-Based Information Fusion for Semi-Supervised Music Genre Classification. *IEEE Transactions on*

Multimedia, *10*(1), 145–152. https://doi.org/10.1109/TMM.2007.911305

Spanias, A., Painter, T., & Atti, V. (2006). *Audio signal processing and coding*. John Wiley & Sons.

Stevens, K. N. (1950). Autocorrelation Analysis of Speech Sounds. *The Journal of the Acoustical Society of America*, *22*(6), 769–771. https://doi.org/10.1121/1.1906687

Sturm, B. L., Santos, J. F., Ben-Tal, O., & Korshunova, I. (2016). *Music transcription modelling and composition using deep learning* (arXiv:1604.08723). arXiv. https://doi.org/10.48550/arXiv.1604.08723

Tzanetakis, G., & Cook, P. (2002). Musical genre classification of audio signals. *IEEE Transactions on Speech and Audio Processing*, *10*(5), 293–302. https://doi.org/10.1109/TSA.2002.800560

Wold, E., Blum, T., Keislar, D., & Wheaten, J. (1996). Content-based classification, search, and retrieval of audio. *IEEE MultiMedia*, *3*(3), 27–36. https://doi.org/10.1109/93.556537

Appendix

This appendix just acts as a placeholder for links to the two digital appendices: the source code repository and the concluding blog post.

- Source code repository: https://github.com/abhishekneerr/Master-Thesis
- Blogpost: https://mct-master.github.io/masters-thesis/2022/06/02-abhishec-thesis-unsupervised-classification.html