AUTOMATIC MUSIC GENRE CLASSIFICATION

RITESH AJOODHA

A dissertation submitted to the Faculty of Science, University of the Witwatersrand, in fulfillment of the requirements for the degree of Master of Science.

Supervised by Dr. Benjamin Rosman and Mr. Richard Klein
Master of Science
School of Computer Science
The University of the Witwatersrand

November 2014

Ritesh Ajoodha: *Automatic Music Genre Classification,* Master of Science.

A dissertation submitted to the Faculty of Science, University of the Witwatersrand, in fulfillment of the requirements for the Degree Master of Science in Computer Science.

SUPERVISORS:
Dr. Benjamin Rosman
Mr. Richard Klein

LOCATION:
Johannesburg, South Africa

"*Music* is the one incorporeal entrance into the higher world of knowledge which comprehends mankind but which mankind cannot comprehend"

— Ludwig van Beethoven [1712 - 1773]

To my parents and brothers

accomplished, unbeatable, visionary

RELATED PAPERS

Some ideas and figures have appeared previously in the following papers:

Ritesh Ajoodha, Richard Klein, and Marija Jakovljevic. *Using Statistical Models and Evolutionary Algorithms in Algorithmic Music Composition.* In Khosrow-Pour Mehdi, editor, *The Encyclopedia of Information Science and Technology.* IGI Global, Hershey, Pennsylvania, United States, 3rd edition, 2014.

Ritesh Ajoodha. *Algorithmic Composition: Using Context-free Grammars, Gaussian Distribution and Evolutionary Algorithms in Algorithmic Music Composition. Honours Research Report.* School of Computer Science, University of the Witwatersrand, Johannesburg, 2013.

"...You see, my dear friend, I am made up of contradictions, and I have reached a very mature age without resting upon anything positive, without having calmed my restless spirit either by religion or philosophy. Undoubtedly I should have gone mad but for music. Music is indeed the most beautiful of all Heaven's gifts to humanity wandering in the darkness. Alone it calms, enlightens, and stills our souls. It is not the straw to which the drowning man clings; but a true friend, refuge, and comforter, for whose sake life is worth living."

— Pyotr Ilyich Tchaikovsky [1840 - 1893]

ACKNOWLEDGEMENTS

I would firstly like to thank my research supervisors and mentors, Dr. Benjamin Rosman and Mr. Richard Klein, for their support far beyond my research undertakings. Dr. Rosman is one of the smartest people I know and has demonstrated and maintained superior research skill and teaching excellence throughout my studies. His abilities to convey knowledge insightfully and concisely with a rich apprehension of language made all of our supervision meetings inspirational and valuable. I hope to one day be able to command an audience with economy and thoroughness as he does as well as being held in high regard by computer science experts around the world.

Richard has been a supportive and inspiring mentor through his interpersonal skills. I am particularly grateful for his support by allowing me to pursue additional projects and studies parallel to my research. His relationship and interactions with students are positively consistent and has a trusted reputation at the school for the courtesy and respect he shows to everyone. Not to mention, his masterful use of technology and broad understanding of computer science, which are both motivating and enriching, particularly whenever I became frustrated with the programming aspects of my research and paid him a surprise visit at his office. Richard demonstrated a strong interest and willingness to always be available when I needed him to be, which I will be most grateful for.

I would also like to thank my *invisible supervisor*, Mr. Mike Mchunu, an associate lecturer at the School of Computer Science at Wits, who is always willing to share with me valuable career and research advice. Whenever I got to the university I would often find little helpful notes and papers relating to my research which Mike would leave behind on my desk. His extensive and ever expanding understanding of machine learning always encouraged me to stop him at any corridor at the university just so I can get an update on the cutting edge machine learning concepts and tools. I am most grateful to him for supporting me through his knowledge, many perceptive conversations and suggestions.

I will forever be thankful to my former research lecturer Professor Marija Jakovjevic. As a lecturer, Prof. Marija demonstrated a love of teaching and a level of commitment necessary to become one of the most accomplished and proud educators of our time. She has provided me with advice many times during post-graduate studies

and remains my role model as a researcher, mentor and teacher. Prof. Marija always encouraged us to think more independently about our experimental results and it is through her that I achieved my first publication. Prof. Marija's motivations is what encouraged me to pursue advanced post-graduate studies in research and I will always be proud to have been one of her students.

Dr. Nishana Parsard's frequent insights about the ins and outs of empirical research are always appreciated. Nishana-mausii is my primary resource for understanding how empirical studies are conducted in a professional and effective way. I treasure and look forward to more moments when you blow my mind by sharing your groundbreaking ideas that will someday change the world. I would also like to thank the loving memory of Chanderman Parsard (1939-2010), who's teachings and guidance will always be missed. I wish he could have lived long enough to see my graduation and brother's wedding.

I dedicate this research towards my mom, dad, and brothers. My parents' tireless efforts to provide a quality education and healthy loving living environments for my brothers and I will always be valued. I cherish all of our family bonding times and know that I would have not made it this far without them. An extra thanks to my mum and dad who helped me furnish this dissertation all in one month. My brothers have been my best friends all my life and I would like to thank them dearly for all their support and tolerance in having to listening to me prattle-on about my research. I thank my beloved grandmother for making me all those delicious study-treats when I battled through the mid-terms and finals. Finally, a special thanks to the newest additions to the family, particularly Meera - my brother's fiancé - for her support and care.

In appreciation to my research funding, I would formally like to acknowledge the funding sources that has made this research possible. I was funded by the *National Research Foundation*; honored by the *Golden Key Chapter Award* for superior scholastic attainment; and finally, given a *Post Graduate Merit Award Scholarship* for Masters and Doctoral Studies.

DECLARATION

I, Ritesh Ajoodha, hereby declare the contents of this masters dissertation to be my own work. This dissertation is submitted for the degree of Master of Science in Computer Science at the University of the Witwatersrand. This work has not been submitted to any other university, or for any other degree.

Johannesburg, South Africa

Ritesh Ajoodha, March 2, 2015

CONTENTS

i INTRODUCTION, REVIEW AND DESIGN 1

1 INTRODUCTION 3
- 1.1 Motivation 3
- 1.2 The Research Problem 5
- 1.3 Research Methodology Overview 7

2 THE FUNDAMENTALS OF MUSICAL ASPECTS 9

3 RELATED WORK 11
- 3.1 Introduction 11
- 3.2 Content-based Feature Extraction 27
  - 3.2.1 Timbre Content-based Features 28
  - 3.2.2 Rhythmic Content-based Features 29
  - 3.2.3 Pitch Content-based Features 29
- 3.3 Related Classification Techniques 30
- 3.4 Music Information Retrieval 32
- 3.5 Contributions 32

4 THE RESEARCH DESIGN 37
- 4.1 Research Design 37
  - 4.1.1 Features 38
  - 4.1.2 Multi-class Classification 40
    - 4.1.2.1 The One-verses-all Paradigm 41
    - 4.1.2.2 The One-verses-one Paradigm 41
  - 4.1.3 GTZAN Dataset 41
  - 4.1.4 Feature Selection 42
    - 4.1.4.1 The Wrapper Method 42
    - 4.1.4.2 The Filter Method 42

ii FEATURE ANALYSIS 43

5 FEATURE REPRESENTATION 45
- 5.1 Introduction 45
- 5.2 Test For Normality 46
  - 5.2.1 Discussion and Conclusion 48
- 5.3 Other Feature Representations 49
- 5.4 Conclusion and Discussion 50

6 MAGNITUDE BASED FEATURES 53
- 6.1 Introduction 53
- 6.2 The Magnitude Spectrum 54
  - 6.2.1 Spectral Slope 55
    - 6.2.1.1 Strongest Frequency 55
  - 6.2.2 Compactness 57
  - 6.2.3 Spectral Decrease 57
  - 6.2.4 Loudness 58
    - 6.2.4.1 Perceptual Sharpness 59
    - 6.2.4.2 Perceptual Spread 59

6.2.5 Onset Detection 60
6.2.6 Octave Band Signal Intensity 60
6.2.7 Peak Detection 60
6.2.7.1 Peak Centroid 60
6.2.7.2 Peak Flux 61
6.2.7.3 Spectral Crest Factor 62
6.2.7.4 Peak Smoothness 63
6.2.8 Spectral Flux 63
6.2.9 Spectral Variability 64
6.3 The Power Cepstrum 64
6.3.1 Mel-Frequency Cepstral Coefficients 67
6.3.2 Flatness 68
6.3.3 Spectral Shape Statistics 69
6.3.3.1 Spectral Centroid 69
6.3.3.2 Spread, Kurtosis and Skewness 71
6.3.4 Spectral Rolloff 72
6.4 Conclusion and Discussion 72
7 TEMPO DETECTION 77
7.1 Introduction 77
7.2 Energy 78
7.2.1 Beat Histogram 79
7.3 Conclusion and Discussion 81
8 PITCH AND SPEECH DETECTION 83
8.1 Introduction 83
8.2 Pitch Detection 83
8.2.1 Amplitude Modulation 83
8.2.2 Zero Crossing Rate 85
8.3 Speech Detection 86
8.3.1 Autocorrelation Coefficients 86
8.4 Envelope Shape Statistics 87
8.5 Conclusion and Discussion 87
9 CHORDAL PROGRESSIONS 89
9.1 Introduction 89
9.1.1 Results 91
9.1.2 Discussion 91
9.2 MFCC-based Chroma vs. Mean-based Chroma 92
9.3 Conclusion 93
III MUSIC GENRE CLASSIFICATION 95
10 AUTOMATIC MUSIC GENRE CLASSIFICATION 97
10.1 Introduction 97
10.2 Information Gain Ranking 97
10.3 Automatic Genre Classification 99
10.4 Conclusion 101
11 CONCLUSION AND FUTURE WORK 103
IV APPENDIX 107
A FUNDAMENTAL MATHEMATICAL CONCEPTS 109
A.1 Root Mean Square 109

A.2 Arithmetic Mean 109
A.3 Geometric Mean 109
A.4 Euclidean Distance 110
A.5 Weighted Mean 110
A.6 Convolution 110
A.7 Complex Conjugate 111
A.8 Hanning Window 111
B ADDITIONAL TABLES AND FIGURES 113
C CLASSIFICATION ALGORITHMS 139
C.0.1 Support Vector Machines 139
C.0.2 Naïve Bayes Classifier 141
C.0.2.1 Introduction 141
C.0.2.2 The Naïve Bayes Classification Process 141
C.0.3 K - Nearest Neighbours 142
BIBLIOGRAPHY 145

LIST OF FIGURES

Figure 1 Music genre classification 8
Figure 2 Research design overview 38
Figure 3 Basic feature extraction process. 39
Figure 4 More detailed feature extraction process. 40
Figure 5 A discrete-time signal. 46
Figure 6 Case processing summary before sample reduction. 47
Figure 7 Case processing summary after sample reduction 47
Figure 8 Energy of all GTZAN genres represented by a Gaussian distribution with $\mu = 0.091$ and the $\sigma = 0.09$. 47
Figure 9 Reduced test for normality. 48
Figure 10 The plot shows the similarity between the empirical cdf of the centered and scaled feature vectors (centroid and rolloff) and the cdf of the standard normal distribution. 48
Figure 11 Normal Q-Q plot for centroid and rolloff features. 49
Figure 12 Confusion matrices for 20-bin feature histogram and mean respectively. 51
Figure 13 The magnitude spectrum of "Arabesque No. 1" by Claude Debussy. 53
Figure 14 Strongest frequency feature values for 10 GTZAN genres using the mean representation. 56
Figure 15 Compactness feature values for 10 GTZAN genres using the mean representation. 58
Figure 16 Peak centroid feature values for 10 GTZAN genres using the mean representation. 61
Figure 17 Peak smoothness feature values for 10 GTZAN genres using the mean representation. 63
Figure 18 Spectral variability feature values for 10 GTZAN genres using the mean representation. 65
Figure 19 Spectral centroid feature values for 10 GTZAN genres using the mean representation. 70
Figure 20 Spectral rolloff feature values for 10 GTZAN genres using the mean representation. 72
Figure 21 Multilayer perceptron classification (79.5%) of GTZAN genres using only magnitude-based features. 74
Figure 22 Energy feature values for 10 GTZAN genres using the mean representation. 78
Figure 23 Strongest beat feature values for 10 GTZAN genres using the mean representation. 80
Figure 24 A tempo-based design matrix using linear logistic regression models classifier achieving 49.3% accuracy. 82
Figure 25 Amplitude modulation of a DTS. 84
Figure 26 Zero crossings feature values for 10 GTZAN genres using the mean representation. 86

Figure 27 Multilayer perceptron classification (67.3%) of GTZAN genres using pitch and speech based features. 88
Figure 28 Global distribution of MFCC-based chroma (3 coefficients) on GTZAN genres. 89
Figure 29 Global distribution of MFCC-based chroma (3 coefficients) on GTZAN's classical, disco, and jazz genres. 90
Figure 30 Confusion matrix using linear logistic regression models on MFCC and chroma-MFCC features. 91
Figure 31 Confusion matrix using linear logistic regression models on MFCC and chroma-MFCC features for 10 GTZAN genres. 92
Figure 32 Information gain ranking of features based on contribution. 98
Figure 33 Relationship between the number of features and classification accuracy using IGR to guide feature trade-off. 100
Figure 34 81% accuracy achieved with linear logistic regression models to classify 10-GTZAN genres using 10-fold cross validation. 100
Figure 35 Description after sample reduction. 135
Figure 36 Empirical cumulative distribution functions for centroid and rolloff. 136
Figure 37 Other examples of probability distributions on spectral centroid. 136
Figure 38 Classification of two datasets using a support vector machine. 140

LIST OF TABLES

Table 1 Related work using GTZAN genres. 12
Table 2 Related work that did not use GTZAN genres. 15
Table 3 List of features and classifiers used by previous authors. 33
Table 4 A general list of features. 39
Table 5 Classification scores for different feature representations for FFT maximum using a variety of classification techniques. 56
Table 6 Classification scores for different feature representations for compactness using a variety of classification techniques. 57
Table 7 Classification scores for different feature representations for peak centroid using a variety of classification techniques. 61
Table 8 Classification scores for different feature representations for peak flux using a variety of classification techniques. 62
Table 9 Classification scores for different feature representations for peak smoothness using a variety of classification techniques. 62
Table 10 Classification scores for different feature representations for spectral flux using a variety of classification techniques. 64
Table 11 Classification scores for different feature representations for spectral variability using a variety of classification techniques. 65
Table 12 Classification scores for different feature representations for MFCCs using a variety of classification techniques. 68
Table 13 Classification scores for different feature representations for spectral centroid using a variety of classification techniques. 70
Table 14 Classification scores for different feature representations for strongest frequency of spectral centroid using a variety of classification techniques. 71
Table 15 Classification scores for different feature representations for spectral rolloff using a variety of classification techniques. 73
Table 16 A list of magnitude-based feature for genre classification. 73
Table 17 Classification scores for different feature representations for energy using a variety of classification techniques. 79
Table 18 Classification scores for different feature representations for beat sum using a variety of classification techniques. 80
Table 19 Classification scores for different feature representations for strongest beat using a variety of classification techniques. 80
Table 20 Tempo-based feature list. 81
Table 21 A list of pitch and speech based features for genre classification. 88
Table 22 Classification scores for chroma on GTZAN genres. 90
Table 23 Classification scores for MFCCs on GTZAN genres. 91
Table 24 Classification scores for both mean-based MFCC and MFCC-based chroma on 10 GTZAN genres. 93
Table 25 Classification scores for both mean-based MFCC and mean-based chroma on 10 GTZAN genres. 93
Table 26 Information gain ranking: features contributions. 99

Table 27 Automatic music genre classification of the thinned features vector. 101
Table 28 Information gain ranking: features contributions. 113
Table 30 Feature list with each feature's representation; number of dimensions; and parameters. 131
Table 31 Parameters of classifiers used. 134
Table 29 Suggested features to discriminate a particular genre. 137

Part I

INTRODUCTION, REVIEW AND DESIGN

Music genre, while often being vaguely specified, is perhaps the most common classification scheme used to distinguish music. Many believe that the fuzziness and instability of genre definitions can lead to inaccurate classification and hence believe that researchers should focus on more sensible methods for improving music browsing or classification over the Internet [McKay and Fujinaga 2004]. While a single human response to genre classification can be biased and stereotypical, there exists a consensus of broad genre definitions across populations worldwide. Genre is positioned between multiple classification methods (e.g. mood or artist) overlaying the Internet. Although these methods are also similarity-based measures across different music meta-data (lyrics, artist, timbre), genre offers a culturally authorised prominence on the construction of traditional classes which is much more functional for music classification.

This part contains four sections: *Introduction*, Chapter 1, where we explore this functionality and emphasise the importance of music genre classification over other approaches; the fundamental aspects of music, Chapter 2, which serves as a brief introduction to the components that together make up music; the *Literature Review*, Chapter 3, where the contributions of other authors are acknowledged and presented; and finally, the *Research Design*, Chapter 4, where the research methodology is presented and the research contributions are realised.

1 INTRODUCTION

The *Internet* has connected the world in numerous ways making information quickly available to us from a universe of data. This is done by using information retrieval techniques. Information retrieval is a cornerstone of information system technologies as retrieving data from a database is needed for countless computer software applications. Such applications include online music databases, for example MusicBrainz [Swartz 2002]. Music databases, such as these, store millions of music files, MusicBrainz for example stores 16 million [Angeles 2009] such music files. Without any sophisticated information retrieval systems, if one wanted to find a particular piece of music in a database, one would have to naïvely compare the desired music piece's *name and artist* with each and every *name and artist* contained in the music database. This task would take a long time to complete, particularly with a large database, and so clever management of the database is needed for quick information retrieval.

1.1 MOTIVATION

The methodology in this dissertation can be used to organise the world of music from classical Baroque to the newly defined Ratchet genre. This is done by studying different approaches to clustering music into genres, where some clusters may overlap depending on whether or not music pieces show diverse characteristics from multiple genres. If one would like to retrieve a piece of music from the database, the genre for the piece should be detected and the piece can be found, if it exists, in the cluster for that genre. This dramatically decreases search time and therefore favours information retrieval speed. Additionally, this approach could be used to generate labels for new music when it is entered into a database. It is hypothesised that this concept can be extended to music video, movie, or even art forms.

Since we are not always given the music name, artist or other meta-data to search by in the music signal itself, we need some element (residing within the music signal) that will aid music organisation. Many authors have questioned the usefulness of genre categorization and whether it should abandoned as it seems to lack real profit for any type of customer. In many music stores, music CDs are arranged by genre, being a natural classification of music, so the customer can easily divert to a particular genre section and consider selecting a CD from a few rows rather than searching the entire CD section of the music store. Analogously, if information in these online music databases are stored by genre, then it is easier to obtain a piece of music by simply searching the online music database for the required genre type. This system could also be able to suggest music based on the customer's genre preference as well. Music has been organised by genre for many years and so customers are already familiar with browsing music within genre categories both on and off the Internet. Lee and

Downie [2004] showed that end-users are more likely to browse by genre than artist similarity, recommendation, or even music similarity.

Large e-commerce companies, such as Amazon, use genre to categorise their media catalogues so consumers can browse through items based on their genre preference. These catalogues are usually labelled manually when inserted into the database. If the media samples are not labelled correctly this will cause the automatic classification algorithm to mis-label music. Other similarity-based approaches, for example mood or composer -based similarity, will suffer the same obscurity between classes as genre classification does simply because of their ground truth. Some examples of shortcomings that are caused by the obscurity of ground truth are the inability to compare classification results with other functional algorithms or reusing components of other classification algorithms that appear successful to build better ones.

One can only begin to understand the value of classification using genre after first understanding the importance of genre to customers. Genre associations are frequently used among consumers to describe and discuss music. If you ask a music lover what *kind* of music does she listen to, the music lover would generally answer in terms of genre than style or mood[1]. Therefore, genre is not only considered by consumers when searching or browsing media data, but hold deeper significance when considering personal aspects of consumers' lives while interacting with other consumers.

Many consumers associate culturally with genre holding their genre preference as personal entities that have a heavy influence on their lives. These genre preferences constantly impact the consumer in a variety of ways psychologically.

Example 1.1. As an example, a listener of an extreme genre such as Grindcore[2], would probably act, talk, and dress differently compared to a listener who enjoys furniture music[3].

North and Hargreaves [1997] showed that music genre has such a pressing influence on consumers that the listener would prefer one song to another based more on the song's genre than the actual song itself. Similar studies by Tekman and Hortacsu [2002] showed that a listener's categorization of genre directly influences whether she will appreciate a piece of music or not.

Successful research in automatic music genre classification contributes considerably to musicology and music theory [McKay and Fujinaga 2005]. Observing interactions between genre classes through content-based features can unveil cultural associations that exist between these classes and is of musicological significance.

1 For example: "I listen to classical and rock music".
2 Considered one of the most caustic sounding genres. Inspired by an amalgam of death metal, crust punk and thrash metal.
3 A very light form of classical music, usually played as background music.

1.2 THE RESEARCH PROBLEM

Music Genre Classification is the process of categorising music pieces using traditional or cultural characteristics. These traditions and cultures are not precisely defined and so over the years it has become vague as to what characteristics secure music to a particular genre or if these bind a piece of music to a clearly defined musical genre at all. According to White [1976], traditional musical aspects are given by four characteristics: melody, harmony, rhythm and sound - where sound is expanded to include timbre, dynamics, and texture (further aspects of music are explained in Chapter 2). These aspects define characteristics of music and can therefore be hypothesised to contribute considerably to the notion of musical genre. As an example, the definition for *R&B* (Rhythm and Blues) and *Rock and Roll* genres according to Oxford [1989] are defined as follows:

> **R&B:** *"A form of popular music of US black origin which arose during the 1940s from blues, with the addition of driving rhythms taken from jazz. It was an immediate precursor of rock and roll."*
>
> **Rock and roll:** *"A type of popular dance music originating in the 1950s, characterised by a heavy beat and simple melodies. Rock and roll was an amalgam of black rhythm and blues and white country music, usually based around a twelve-bar structure and an instrumentation of guitar, double bass, and drums."*

Although these definitions refer loosely to rhythm and instrumentation (timbre), there is no specification for dynamics, texture, structure, pitch, tempo, harmony or scale-use, which according to White [1976] and Owen [2000] are also fundamental aspects that define music. Therefore, these definitions lack clarity and rather refer more to the connections between genres, which are less useful (e.g.*"from blues, ...taken from jazz"* or *"popular dance,... black rhythm and blues and white country music"*). Also, since these definitions are qualitative they come across as subjective and therefore are difficult to automate. It is seen that these genre definitions refer recursively to other genres, making them context dependent. Furthermore, many artists do not abide by "genre definitions", even though the composer might have been inspired by one, which makes us question whether or not some composers are accepted by currently "defined" music genres. For this reason music genre classification is categorised using human discretion and is therefore prone to errors and subjectivity as many pieces of music sit on boundaries between genres [Li *et al.* 2003].

Successful genre classification makes use of cultural-based rather than content-based feature dissimilarity between genre classes. Consequently, in order to perform genre classification optimally, meta-data that describes cultural aspects needs to be extracted somewhere within some appendix of the signal or from the Internet. Although some of this meta-data might be already available, there is little motivation to keep extensive records of cultural attributes for every piece of music on the Internet or as an appendix to the music recording[4] itself. Acquiring these cultural-based

4 Perhaps as an attachment or as an appendix of information about cultural features.

features for music genre classification could unlock the potential of this research to excel through content-based feature constraints.

Constructing dependable ground truth for classifiers is a significant component for supervised learning algorithms, and hence for successful automatic music genre classification. The fuzziness of genre definitions can cause some music to become difficult to classify under one label. This is because certain features of a piece of music will suggest one genre while other features might suggest another similar genre or even multiple other genres. It is through this ambiguity that the limitations of the ground truth becomes inescapably bounded as many people may disagree on a particular genre classification, limiting the full potential of this research. Additionally, many genres do not only sound similar but also hold many sub-genres which share some similar characteristics.

Example 1.2. Western classical music consists of 3 broad periods: *Early*; *Common practice* and *Modern & Contemporary*. *Early* contains 2 sub-genres[5]; *Common Practice* contains 3 sub-genres[6]; and *Modern & Contemporary* contains 4 sub-genres[7]. These 9 broad sub-genres contain in themselves other music genres which have their own features. Using this framework the *Violin Partita No.2* by *Johann Sebastian Bach*, which is an early *Baroque* piece should be classified into a different genre than *Deux arabesques* by *Claude Debussy* which is an early *Impressionism*. Again, every Western classical composers' music is further classified by instrumentation which is also a form of genre. For example, *Claude Debussy* composed music from eight different music genres that included *Orchestral*; *Ballet*; *Soloist and orchestra*; *Chamber*; *Solo piano*; *Piano four hands or two pianos*; *Voice and piano*; and *Other vocal* which are just a small extension of western classical music. The difficulty of genre classification increases tremendously when considering hundreds of other genre types and their respective sub-genres. This now creates another issue as manual classification is not only subjective but now a single piece of music can belong to possibly hundreds of similar genre types which poses a bigger conundrum, this expresses the difficulty of this problem.

Very little experimental research has been done on human responses to music genre classification. A study done by Gjerdingen and Perrott [2008] who conducted a survey in which participants were given brief excerpts of commonly recorded music from one of ten broad genres of music. The participants were asked to categorise the music excerpts into one of ten music genre labels. Gjerdingen and Perrott [2008]; Perrot and Gjerdigen [1999] discovered that humans with little to moderate music training can achieve about 70% classification accuracy, based on 300ms of audio per recording. One may argue that had the participants been exposed to more of the recording perhaps they could have obtained better genre classification results. Another study by Lippens *et al.* [2004] considered 27 human listeners where each was given 30 seconds of popular music and was asked to categorise the pieces in one of six musical genre labels, where one of these labels was "Other". Listeners might have labelled recordings as "Other" not because the recording belonged there, but because the genre was

5 Medieval (500-1400) and Renaissance (1400-1600)
6 Baroque (1600-1760); Classical (1730-1820); and Romantic (1780-1910)
7 Modern (1890-1930); 20^{th} century (1901-2000); Contemporary (1975-present); and 21^{st} century (2001-present)

ambiguous. This could also have occurred because of the uncertainty of the listener trying to meet the requirements of a defined sub-genre. The listeners achieved an inter-participant genre agreement rate of 76% only. Although Gjerdingen and Perrott [2008] and Lippens *et al.* [2004] provide an awareness of genre classification performance bounds imposed by human responses to genre classification, further study in experimental research is needed to draw more concise conclusions regarding human responses to genre classification and how this affects ground truth.

From these results it is seen that humans are biased and subjective in genre classification, which ultimately leads to a lack of consensus in genre labels and thus poor quality of ground truth. Although there exists more authoritative ground truth sources (e.g. such as AllMusic, Gracenote or Discogs), that contain information about style and mood, these online music guide services tend to bias genre labels to entire albums rather than for each individual song - even though it is well known that a single album can contain multiple genre types depending on each recording. The methods to how genre labels were obtained for albums and recordings are not presented, which makes one question if these media tags are reliable and consistent. In addition, doing genre classification by hand is a monotonous and uninteresting task and so when presented with 16 000 000 music files to label, such as in MusicBrainz, humans are likely to take even longer to label each piece by genre resulting in further imprecision, not to mention that this task demands time and rare expertise to perform successfully. Assured genre labelling is needed to prevent over-training supervised classification models with large training samples as genre detection models need to integrate the same type of intricacy that humans achieve with the delicate boundaries between genre labels.

To make matters worse, genre definitions evolve over time and continuously give rise to newer structures that have significantly different feature compositions. These newer genre might exhibit qualities that the learning model might fail to detect or perhaps a recently evolved genre might have a completely different definition as it had many years ago. These changes in definition will significantly alter the integrity of the ground truth. This will cause the learning model to lose credibility over time and will have to be retrained periodically. Therefore, regardless of feature dimensionality, well-built classification procedures are required to classify features successfully with some regard for genre development. This is difficult as the scalability of even the most powerful supervised classification models are unsatisfactory [McKay and Fujinaga 2006].

A review of the literature shows very few capable genre classification systems. For this reason, systems thus far have not adopted automatic genre classification models for media retrieval and recommendation. Noteworthy genre classification rates include Sturm [2013b] who achieved 73-83% on 10 genres and Bergstra *et al.* [2006] who achieved 82.50% on 10 GTZAN (a music dataset by George Tzanetakis) genres. This work aims to develop single genre classification methods for the purpose of database retrieval and recommendation systems.

1.3 RESEARCH METHODOLOGY OVERVIEW

Music pieces are represented as discrete-time signals (DTSs). Genre classification from a collection of DTSs requires the extraction of signal features and subsequently

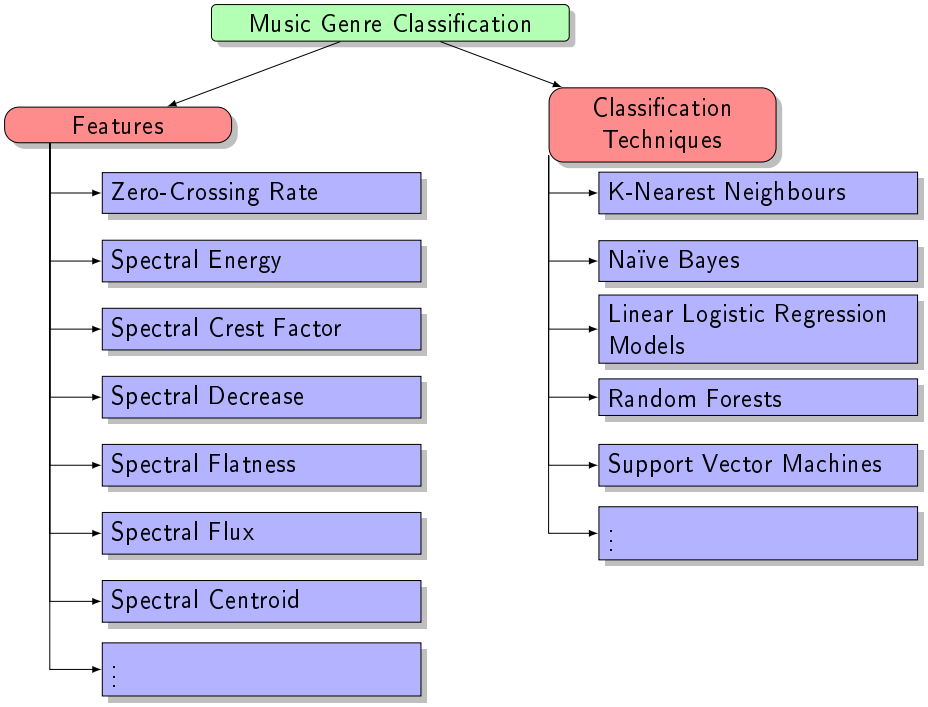


Figure 1: Music genre classification

using a classification algorithm. These features act as individual descriptors for the DTS and so a range of different features (e.g. zero-crossing rate, energy, crest factor etc.) are needed to thoroughly describe it. After successfully performing a feature extraction, a design matrix is created which contains all of the descriptions necessary for every DTS in the collection. Lastly the design matrix is modelled in n dimensions[8] using a set of classification techniques such as K-nearest neighbours, neural networks, and support vector machines. Figure 1 shows a list of classification approaches and feature examples used to classify and describe the data respectively for music genre classification. Chapter 2 introduces the reader to fundamental aspects of music. These aspects, if understood well, can serve as foundational knowledge when understanding the features described in Part ii.

8 Where n is the number of features.

2

THE FUNDAMENTALS OF MUSICAL ASPECTS

Any part, feature, dimension, element or characteristic of musical audio is considered as an aspect of music. Musical aspects are fundamental factors used to describe music and include, but are not limited to: pitch, tempo, rhythm, melody, harmony, voice, timbre, expression, and structure. In addition to these fundamental aspects of music there are also other important aspects that are employed like scale-use and phrasing [Ajoodha *et al.* 2014]. White [1976] proposed six traditional musicological aspects:

- **Melody** is a set of musical notes played meaningfully in succession.
- **Harmony** is the use of simultaneous pitches, or chords [Malm *et al.* 1967].
- **Rhythm** is the observation of periodically occurring strong and weak beats that are contained within a melody [Dictionary 1971].
- **Tone** is a steady periodic sound. A musical tone is characterised by its duration, pitch, intensity (or loudness), and quality [Roederer and Roederer 1995].
- **Form** describes the layout or general structure of a composition [Schmidt-Jones 2011; Brandt *et al.* 2012]. This general structure is noted by the division of different passages or sections that exist in the music. These passages or sections are considered the same if there is a large degree of similarity between them. Examples of musical form include binary form, strophic form, and rondo form.
- **Tempo** is how fast or slow a piece is played and can be measured as the number of beats per minute [Ajoodha *et al.* 2014].

Owen [2000] proposed a more grounded definition of sound over that originally suggested by White [1976]. Owen [2000] included the following aspects of music alongside White [1976]:

- **Pitch** refers to an organisation of notes based of their frequency [Klapuri and Davy 2006]. Through this organisation of frequency some notes are considered as high or low relative to other notes [Plack *et al.* 2006], this type of organisation is best achieved in stable and clear sound distinguished from noise [Randel 2003]. It is understood that pitch, along with duration, timbre, and dynamics, are the chief components when humans observe sound [Patterson *et al.* 2010].
- **Dynamics** are local and global audio volumes (loudness).
- **Timbre** of a musical sound is the distinct quality of sound that exists for every instrument. The unique sound distinguishes different types of sound productions. Examples of differing timbres are obvious if one considers the same note from a voice, a string instrument and a percussion instrument – such as an opera singer, a violin and a piano.

While the proposed *primary* aspects by White [1976] and Owen [2000] form a cornerstone in most music definitions, the permutations of these *primary* aspects must be considered. Combining these *primary* aspects to certain degrees yields *secondary* aspects of music like structure, texture, style, and aesthetics. These *secondary* aspects encompass the following:

- **Structure** refers to global aspects of a music piece. These include phrasing, period, repetition, variation, development, form, etc.
- **Texture** is how the primary features pitch and timbre correlate. It includes: homophony, where two or more musical sounds move together (usually in chord progressions); polyphony, which describes how two or more melodies move together, as opposed to *monophony*, where a single melody with no harmony is presented; heterophony, where multiple monophonic lines (from the same theme) are played together; and simultaneity, where multiple musical textures occur together and not in succession.
- **Style** is defined as the way musical aspects are used. Style is used to classify composers, bands, period, region, or interpretation. For example: using non-fluctuating dynamics, strict polyphony, variation, and repetition could result in a baroque genre. A genre can be seen as a cultural classification of similar styles of music.
- **Aesthetics or mood** are how the music influences the listener's emotions. For example: a slow and yearning melody may make you sad or depressed, while a lively piece might cause you to be happy or energetic.

3

RELATED WORK

3.1 INTRODUCTION

There exist several techniques to perform *feature extraction* for *music genre classification* used in *music information retrieval*, however the most prevalent of these techniques are that of:

1. *reference feature extraction*, where the composer and other meta-data[1] are extracted;
2. *content-based acoustic feature extraction*, where timbre features are examined for extraction (e.g. pitch, beat, rhythm or tonality);
3. *symbolic feature extraction*, where features are extracted directly from the music score;
4. and *text-based feature extraction*, where features are extracted from the lyrics of a piece.

This research employs *content-based acoustic feature extraction techniques* as the independence from the meta-data needed to perform the other above feature extraction methods is rewarding. Context-based acoustic feature extraction techniques rely only on the signal itself and has proven a potential solution to genre classification problems. Therefore, the contributions and recent work in this area must be presented.

In Figure 1, it is seen that two major requirements must be addressed to perform general music classification: a choice of *features*, and a choice of *classification algorithm*. This chapter presents methodologies[2] already explored by other authors who have made significant contributions in music genre classification as well as other prevalent audio analysis fields (e.g. from speech recognition to signal compression).

Since mature descriptive techniques for audio feature extraction became available, in the 1990's, the domain of content-based music information retrieval experienced a major uplift [Lidy and Rauber 2005], in particular the study of music genre classification [Fu *et al.* 2011]. Aside from the numerous developments that music genre classification can have on music information retrieval, there are many other applications upon the success or development over this problem, for example music auto-tagging[3] [Casey *et al.* 2008] and recommendation. As a result of this, *content-based feature extraction techniques* has attracted a lot of attention as researches wish to exploit

1 Those that do not require extended work to obtain, e.g. the title of the piece or perhaps the composers name.
2 Which comprise of features and classification algorithms.
3 Genre auto-tagging offers little data-discrepancies and eliminates errors.

content-based feature extraction methods to maximise classification accuracy, and by doing so, maximise music information retrieval speed. Current approaches hypothesise the usefulness of some features, and classification results deem their accuracy. Music genre has been favored by many researchers as a hypothetical true descriptor of musical content [Aucouturier and Pachet 2003] that can be used to categorise music into clusters - even though genre is not clearly defined as, stated in Section 1.2, the components of genre definitions are unlimited [White 1976] and fuzzy [Scaringella *et al.* 2006]. Optimistically, this research uses background information and previous success [Scaringella *et al.* 2006; Tzanetakis and Cook 2002; Owen 2000], as seen in Table 1 and Table 2, to assume that audio signals contain some details describing the inexplicit definitions of musical genre. Table 1 shows related work using GTZAN genres for music genre classification and Table 2 presents related work using other dataset such as MIREX 2005, ISMIR 2004, and other corpus constructions.

Remark 3.1. An accepted protocol for music genre classification is by using the *bag-of-features (BOF)* approach [Scaringella *et al.* 2006; McFee *et al.* 2012; Tzanetakis and Cook 2002; Owen 2000], where audio signals are modelled by their long-term statistical distribution of their short-term spectral features [Panagakis *et al.* 2010].

Content-based acoustic features are summarised by the following conventional extension: *timbre content features*; *rhythmic content features*; *pitch content features*; or their combinations [Tzanetakis and Cook 2002]. Section 3.2 describes content-based related work while Section 3.2.1, Section 3.2.2, and Section 3.2.3 provide related work using *timbre*, *rhythm*, and *pitch feature extraction* respectively.

As shown in Figure 1, the classification step is just as important as the feature extraction step, and so paramount related work in this step must be realised. Section 3.3 presents related work on classification algorithms used for music genre classification. Although there has been much work in this field, only work directly related to content-based classification will be presented to express the novelty of this research. In order to compare the results of this study with those of other studies, orthodox datasets are used to make the reported classification accuracies comparable (Section 4.1.3). Noteworthy music genre classification model accuracies are given by Table 1 and Table 2, which suggest that *content-based feature extraction using BOF techniques* together with classifier tools such as: k-nearest-Neighbour (k-NN); the support vector machines (SVM); Gaussian mixture model (GMM); linear discriminant analysis (LDA); non-negative matrix factorisation; and non-negative tensor factorisation, have been proven most powerful.

Table 1: Related work using GTZAN genres.

Noteworthy Genre Classification Approaches and Contributions
(1) Benetos and Kotropoulos [2008]
Feature set: Audio power, audio fundamental frequency, total loudness, specific loudness, audio spectrum centroid, spectrum rolloff frequency, audio spectrum spread, audio spectrum flatness, Mel-frequency cepstral coefficients, autocorrelation values, log attack time, temporal centroid, zero-crossing rate.

Continued on next page

Table 1 – *Continued from previous page*

Dataset: GTZAN	
Classification Algorithm	**Accuracy**
Non-negative tensor factorisation (NTF)	75.00%
Non-negative matrix factorisation (NMF)	62.00%
Local NMF (LNMF)	64.33%
Sparse NMF (SNMF) 1 ($\lambda = 0.1$)	64.66%
SNMF 2 ($\lambda = 0.001$)	66.66%
Multi-layer Perceptrons (MLP)	72.00%
Support Vector Machines (SVM)	73.00%
(2) Bergstra *et al.* [2006]	
Feature set: Fast Fourier transform coefficients, real cepstral coefficients, Mel-frequency cepstral coefficients, zero-crossing rate, spectral spread, spectral centroid, spectral rolloff, autoregression coefficients.	
Dataset: GTZAN	
Classification Algorithm	**Accuracy**
ADABOOST	82.50%
(3) Cast *et al.* [2014]	
Feature set: Mel-frequency cepstral coefficients.	
Dataset: GTZAN, The authors used their model to classify four or three of the following genres: classical, metal, pop, and country.	
Classification Algorithm	**Accuracy**
Naïve Bayes (3-genres)	98.33%
k-Means (3-genres)	95.00%
k-Medoids (3-genres)	87.00%
k-nearest neighbours (3-genres)	86.67%
Naïve Bayes (4-genres)	93.75%
k-Means (4-genres)	85.00%
k-Medoids (4-genres)	80.00%
k-nearest neighbours (4-genres)	72.50%
(4) Holzapfel and Stylianou [2008]	
Feature set: MFCCs, non-negative matrix factorisation based features.	
Dataset: GTZAN, Five fold cross validation.	
Classification Algorithm	**Accuracy**
Gaussian Mixture Models (NMF 5)	71.70%
Gaussian Mixture Models (NMF 10)	74.00%
Gaussian Mixture Models (NMF 15)	73.90%

Continued on next page

Table 1 – *Continued from previous page*

Gaussian Mixture Models (NMF 20)	73.20%
Gaussian Mixture Models (MFCC 10)	70.30%
Gaussian Mixture Models (MFCC 20)	71.60%
Gaussian Mixture Models (MFCC 30)	73.00%
Gaussian Mixture Models (MFCC 40)	72.30%

(5) Li *et al.* [2003]

Feature set: MFCCs, spectral centroid, spectral rolloff, spectral flux, zero crossings, low energy, rhythmic content features, pitch content features.

Dataset: GTZAN

	SVM1	SVM2	LDA	KNN
{Bl, Cl}	98.00%	98.00%	99.00%	97.50%
{Bl, Cl, Co}	92.33%	92.67%	94.00%	87.00%
{Bl, Cl, Co, Jaz}	90.50%	90.00%	89.25%	83.75%
{Bl, Cl, Co, Jaz, Met}	88.00%	86.80%	86.20%	78.00%
{Bl, Cl, Co, Jaz, Met, po}	84.83%	86.67%	82.83%	73.50%
{Bl, Cl, Co, Jaz, Met, po, hi}	83.86%	84.43%	81.00%	73.29%
{Bl, Cl, Co, Jaz, Met, po, hi, Reg}	81.50%	83.00%	79.13%	69.38%
{Bl, Cl, Co, Jaz, Met, po, hi, Reg, Ro}	78.11%	79.78%	74.47%	65.56%

(6) Lidy *et al.* [2007]

Feature set: Loudness, amplitude modulation, rhythm histogram, statistical spectrum descriptor, onset features, and symbolic features.

Dataset: GTZAN

Classification Algorithm	**Accuracy**
linear Support Vector Machines	76.80%

(7) Panagakis *et al.* [2008]

Feature set: Multiscale spectro-temporal modulation features

Dataset: GTZAN

Classification Algorithm	**Accuracy**
Non-Negative Tensor Factorization	78.20%
High Order Singular Value Decomposition	77.90%
Multilinear Principal Component Analysis	75.01 %

(8) Sturm [2013b]

Continued on next page

Table 1 – *Continued from previous page*

Feature set: The system of (SRCRP) uses six short-term features: octave-based spectral contrast (OSC); Mel-frequency cepstral coefficients (MFCCs); spectral centroid, rolloff, and flux; zero-crossings; and four longterm features: octave-based modulation spectral contrast (OMSC): "low-energy"; modulation spectral flatness measure (MSFM); and modulation spectral crest measure (MSCM).	
Dataset: GTZAN	
Classification Algorithm	**Accuracy**
Quadratic discriminate classifier with sparse approximation	78 – 83%
(9) Tzanetakis and Cook [2002]	
Feature set: Spectral centroid, spectral rolloff, spectral flux, first five Mel-Frequency sepstral coefficients, low-energy feature, zero-crossing rate, means and variances, beat histogram	
Dataset: GTZAN, 10-fold cross-validation	
Classification Algorithm	**Accuracy**
Gaussian Mixture Models	61.00%
K-nearest neighbour	60.00%

Table 2: Related work that did not use GTZAN genres.

Noteworthy Genre Classification Approaches and Contributions	
(1) Basili *et al.* [2004]	
Feature set: Intervals, instruments, instrument classes, time changes, note extension.	
Dataset: Corpus construction, 300 MIDIs, 5-fold cross-validation.	
Classification Algorithm	**Accuracy**
Naïve Bayes (NB)	62.00%
Voting Feature Intervals (VFI)	56.00%
J48 - Quianlan algorithm	58.00%
Nearest neighbours (NN)	59.00%
RIPPER (JRip)	52.00%
(2) Berenzweig *et al.* [2003]	
Feature set: Mel-frequency cepstral coefficients.	
Dataset: For these experiments, the authors simply hand-picked "canonical" artists and genres with a criterion in mind.	
Classification Algorithm	**Accuracy**

Continued on next page

Table 2 – *Continued from previous page*

Gaussian Mixture Models	62.00%
(3) Bergstra *et al.* [2006]	
Feature set: Fast Fourier transform coefficients, real cepstral coefficients, Mel-frequency cepstral coefficients, zero-crossing rate, spectral spread, spectral centroid, spectral rolloff, autoregression coefficients.	
Dataset: MIREX2005	
Classification Algorithm	**Accuracy**
ADABOOST	82.34%
(4) Cataltepe *et al.* [2007]	
Feature set: Timbral features: spectral centroid, spectral rolloff, spectral flux, time-domain zero crossing, low energy, Mel-frequency spectral coefficients, Means and variances of the spectral centroid, spectral rolloff, spectral flux, zero crossing (8 features), and low energy (1 feature) results in 9-dimensional feature vector and represented in experimental results as STFT label. Rhythmic and beat features: BEAT (6 features), STFT (9 features), MFCC (10 features), MPITCH (5 features), ALL (30 features).	
Dataset: McKay and Fujinaga's 3-root and 9-leaf genre data set (3 genres: Classic, jazz, pop)	
Classification Algorithm	**Accuracy**
10-Nearest Neighbours	75.00%
Linear discriminate analysis	86.33%
(5) Cilibrasi *et al.* [2004]	
Feature set: Similarity function between pieces.	
Dataset: Corpus construction, the following three genres were used: 12 rock pieces, 12 classical pieces (by Bach, Chopin, and Debussy), and 12 jazz pieces.	
Classification Algorithm	**Accuracy**
Distanced Based Classification	89.50%
(6) Dannenberg *et al.* [1997]	
Feature set: 13 low-level features based on the MIDI data: averages and standard deviations of MIDI key number, duration, duty factor, pitch and volume, counts of notes, pitch bend messages, and volume change messages.	
Dataset: The performer watches a computer screen for instructions. Every fifteen seconds, a new style is displayed, and the performer performs in that style until the next style is displayed - 25 examples each of 8 styles.	
Classification Algorithm	**Accuracy**
Bayesian Classifier (4-classes)	98.10%
Linear Classifier (4-classes)	99.40%
Neural Networks (4-classes)	98.50%
Bayesian Classifier (8-classes)	90.00%

Continued on next page

Table 2 – *Continued from previous page*

Linear Classifier (8-classes)	84.30%
Neural Networks (8-classes)	77.00%
(7) Deshpande *et al.* [2001]	
Feature set: Spectrograms and Mel-frequency spectral coefficients.	
Dataset: 157 song samples were collected from the Internet. From each of those, a 20 second long clip was extracted. These 20 sec long clips were used throughout experiments, both for training and testing - 3 genres: rock, classical and jazz.	
Classification Algorithm	**Accuracy**
k-Nearest Neighbours	75.00%
Gaussian Mixture Models	Failed (The distribution was not Gaussian)
Support Vector Machines (classical and non-classical genres)	90.00%
(8) Dixon *et al.* [2004]	
Feature set: Rhythmic features: none, periodicity histograms, IOI histograms, periodicity & IOI hist, tempo attributes, and all (plus bar length).	
Dataset: The authors collected 698 samples of standard and Latin ballroom dance music [bal], each consisting of approximately the first 30 seconds of a piece. The music covers the following eight classes: Cha Cha, Jive, Quickstep, Rumba, Samba, Tango, Viennese Waltz and (slow) Waltz.	
Classification Algorithm	**Accuracy**
AdaBoost (No rhythmic descriptors, With rhythmic patterns)	50.10%
AdaBoost (11 periodicity hist., With rhythmic patterns)	68.10%
AdaBoost (64 IOI hist., With rhythmic patterns)	83.40%
AdaBoost (75 Periodicity & IOI hist., With rhythmic patterns)	85.70%
AdaBoost (Tempo attributes, With rhythmic patterns)	87.10%
AdaBoost (All (plus bar length), With rhythmic patterns)	96.00%
AdaBoost (No rhythmic descriptors, Without rhythmic patterns)	15.90%
AdaBoost (11 periodicity hist., Without rhythmic patterns)	59.90%
AdaBoost (64 IOI hist., Without rhythmic patterns)	80.80%
AdaBoost (75 Periodicity & IOI hist., Without rhythmic patterns)	82.20%

Continued on next page

Table 2 – *Continued from previous page*

AdaBoost (Tempo attributes, Without rhythmic patterns)	84.40%
AdaBoost (All (plus bar length), Without rhythmic patterns)	95.10%

(9) Gouyon *et al.* [2000]

Feature set: Attack time, decay time, envelope of the attack region, the time difference between the index of maximum slope and the onset[4], number of sinusoids in the Prony modelling of the reversed attack and decay region, maximum magnitude component in the Prony modeling of the reversed attack and decay region, exponential decay factor of the maximum magnitude component in the Prony modeling of the reversed attack and decay region, maximum magnitude component in the Fourier transform of the attack and decay region - below the Strongest Partial FFT Decay, maximum magnitude component in the Fourier transform of the whole percussive sound, Local mean energy of the attack and decay region, Local mean energy of the whole percussive sound, proportion between local mean energy of the, attack and the decay regions, zero-crossing Rate of the attack region - below the ZCR Attack, ZCR of the decay region - below the ZCR decay, ZCR of the whole percussive sound.

Dataset: The authors selected percussive sounds most suited to the problem. They considered a database consisting of samples taken from the Korg 05RW's general MIDI drum kit. These sounds are classified into two categories by hand: bass drum sounds (15 sounds) and snare sounds (6 sounds).

Classification Algorithm	Accuracy
Agglomerative Clustering (2-classes)	87.50%

(10) Grimaldi *et al.* [2003]

Feature set: 48 time-frequency features

Dataset: 200 instances divided in 5 different musical genres (Jazz, Classical, Rock, Heavy Metal and Techno), with 40 items in each genre. Each item is sampled at 44100 Hz, mono. 10-fold cross validation

Classification Algorithm	Accuracy
Simple K nearest neighbour (with No FS feature ranking procedure)	78.50%

(11) Guo and Li [2003]

Feature set: Total spectrum power, sub-band powers, brightness, bandwidth, pitch frequency, Mel-frequency cepstral coefficients. The means and standard deviations of the L MFCCs are also calculated over the nonsilent frames, giving a 2L-dimensional cepstral feature vector, named "Ceps". The Perc and Ceps feature sets are weighted and then concatenated into still another feature set, named "Perc-Ceps", of dimension 18+2L.

Continued on next page

4 That gives an indication of the sharpness or the smoothness of the attack.

Table 2 – *Continued from previous page*

Dataset: An audio database of 409 sounds from Muscle Fish is used for the experiments, which is classified into 16 classes by Muscle Fish.	
Classification Algorithm	**Accuracy**
Support Vector Machines (Ceps10)	14.65% error rate (29)
k-nearest neighbours (Ceps40)	19.70% error rate (39)
5- nearest neighbours (Ceps40)	24.24% error rate (40)
Nearest center (Ceps80)	40.40% error rate (80)
Support Vector Machines (Ceps8)	08.08% error rate (16)
k-nearest neighbours (Ceps8)	13.13% error rate (26)
5- nearest neighbours (Ceps80)	32.32% error rate (41)
Nearest center (Ceps60)	20.71% error rate (64)
(12) Hamel *et al.* [2011]	
Feature set: Mel-scaled energy bands, octave-based spectral contrast features, spectral energy bands, feature learning and deep Learning using neural networks	
Dataset: MIREX 2009, Magnatagatune dataset consists of 29-second clips with annotations that were collected using an online game called TagATune.	
Classification Algorithm	**Accuracy**
mel-spectrum + Pooled Features Classifier (Average AUC-Tag)	82.00%
Principal Mel-Spectrum Components + Pooled Features Classifier (Average AUC-Tag)	84.50%
Principal Mel-Spectrum Components + Multi-Time-Scale Learning model (Average AUC-Tag)	86.10%
mel-spectrum + Pooled Features Classifier (Average AUC-Clip)	93.00%
Principal Mel-Spectrum Components + Pooled Features Classifier (Average AUC-Clip)	93.38%
Principal Mel-Spectrum Components + Multi-Time-Scale Learning model (Average AUC-Clip)	94.30%
(13) Holzapfel and Stylianou [2008]	
Feature set: MFCCs, non-negative matrix factorisation based features.	
Dataset: ISMIR2004, 5-fold cross validation.	
Classification Algorithm	**Accuracy**
Gaussian Mixture Models (NMF 5)	75.70%
Gaussian Mixture Models (NMF 10)	83.50%

Continued on next page

Table 2 – *Continued from previous page*

Gaussian Mixture Models (NMF 15)	77.70%
Gaussian Mixture Models (NMF 20)	78.60%
Gaussian Mixture Models (MFCC 10)	60.00%
Gaussian Mixture Models (MFCC 20)	61.10%
Gaussian Mixture Models (MFCC 30)	67.70%
Gaussian Mixture Models (MFCC 40)	67.30%

(14) Jiang *et al.* [2002]

Feature set: Octave-based spectral contrast. The mean and standard deviation of spectral contrast composes a 24-dimension feature for a music clip.

Dataset: "There are about 1500 pieces of music in our database for experiments, and five music types are included baroque music, romantic music, pop songs, jazz, and rock. Most of the baroque pieces in thc database are literatures of Bach and Handel, who are the most important composers in the baroque era. The romantic database is composed of literatures of Chopin, Schubert, Liszt. Beethoven, and other composers in the romantic era. Pop songs are those singed by some popular singers, which includes nine men and sixteen women. Jazz and rock in the database also include literatures of many different composers. In each music type database, different possible musical form and musical instruments are included. All the music data in the database are 16kHz. 16 bits, mono wave files. About 6250 IO-second clips, which are randomly selected from the IS00 pieces of music, compose the classification database, where 5000 is for training and 1250 for testing. For each music type, there are about 1000 clips in the training set, and about 250 clips in the testing set. IO-second clips from the same music piece would not appear both in the training set and lesting set. In the classification experiments on whole music, the training data is the same as those for 10- second music clips, while the testing data is composed by the music piece whose clips are presented in the original testing data set."

Classification Algorithm	Accuracy
Gaussian Mixture Models	82.30%

(15) Lambrou *et al.* [1998]

Feature set: Mean, variance, skewness, kurtosis, accuracy, angular second moment, correlation, and Entropy.

Dataset: (12) musical signals (4 Rock, 4 Piano, and 4 Jazz), for the training stage of the classification procedure.

Classification Algorithm	Accuracy
Least Squares Minimum Distance Classifier (Kurtosis vs. Entropy)	91.67%

(16) Lee *et al.* [2007]

Feature set: MFCCs, octave-based spectral contrast, octave-based modulation spectral contrast (OMSC).

Continued on next page

Table 2 – *Continued from previous page*

Dataset: In the experiments, there are 1783 music tracks derived from compact disks. All music tracks in our database are 44.1 kHz, 16 bits, stereo wave files. Half of the music tracks are used for training and the others for testing. All the music tracks are classified into seven classes including 342 tracks of chamber (Ch), 405 tracks of dance (D), 183 tracks of hip-hop (H), 203 tracks of jazz (J), 178 tracks of orchestra (O), 201 tracks of popular (Po), and 271 tracks of rock (R) music.

Classification Algorithm	Accuracy
Nearest Neighbour	84.30%

(17) Li *et al.* [2001]

Feature set: Average energy, spectral centroid, spectral bandwidth, spectral rolloff, band ratio, delta magnitude, zero-crossing rate, pitch, Mel-frequency cepstral coefficients, LPC, delta.

Dataset: The authors collected a large number of audio clips with 7 categories: Noise, speech, music, speech + noise, speech + speech, speech music. Data was collected from TV programs, talk shows, news, football games, weather reports, advertisements, soap operas, movies, late shows, etc.

Classification Algorithm	Accuracy
Bayesian Classifier under the assumption that each category has a multidimensional Gaussian distribution	90.10%

(18) Li *et al.* [2003]

Feature set: MFCCs, spectral centroid, spectral rolloff, spectral flux, zero crossings, low energy, rhythmic content features, pitch content features.

Dataset: "The collection of 756 sound files was created from 189 music albums as follows: From each album the first four music tracks were chosen (three tracks from albums with only three music tracks). Then from each music track the sound signals over a period of 30 seconds after the initial 30 seconds were extracted in MP3."

	SVM1	SVM2	LDA	KNN
{DWCHs}	71.48%	74.21%	65.74%	61.84%
{Beat+FFT+MFCC+Pitch}	68.65%	69.19%	66.00%	60.59%
{FFT+MFCC}	66.67%	70.63%	65.35%	60.78%
{Beat}	43.37%	44.52%	40.87%	41.27%
{FFT}	61.65%	62.19%	57.94%	57.42%
{MFCC}	60.45%	67.46%	59.26%	59.93%
{Pitch}	37.56%	39.37%	37.82%	38.89%

(19) Lidy and Rauber [2005]

Continued on next page

Table 2 – *Continued from previous page*

Feature set: Fast Fourier transform (FFT) with hanning window function (23 ms windows) and 50% overlap, bark scale, spectrum energy, loudness, specific loudness, amplitude modulation (0 to 43 Hz) - 0 through 10 Hz is considered in the rhythm patterns, weight modulation amplitudes, statistical spectrum descriptor, rhythm histogram features.	
Dataset: ISMIR2004, 10-fold cross validation.	
Classification Algorithm	**Accuracy**
Support Vector Machines	79.70%
(20) Lidy *et al.* [2007]	
Feature set: Loudness, amplitude modulation, rhythm histogram, statistical spectrum descriptor, onset features, and symbolic features.	
Dataset: MIREX2007	
Classification Algorithm	**Accuracy**
linear Support Vector Machines	75.57%
(21) Mandel and Ellis [2005]	
Feature set: MFCCs.	
Dataset: Uspop2002 collection [Berenzweig *et al.* 2004; Ellis *et al.*].	
Classification Algorithm	**Accuracy**
Support Vector Machine	68.70%
(22) Mandel and Ellis [2007]	
Feature set: Temporal Pipeline: Mel spectrum → Magnitude Bands → Low Freq Modulation → Envelope Cepstrum → Temporal features → Combined features with Spectral pipeline → final features. Spectral pipeline: Mel spectrum → MFCCs → Covariance → Spectra features → Combined features with temporal pipeline → final features.	
Dataset: MIREX2007	
Classification Algorithm	**Accuracy**
Support Vector Machines	75.03 %
(23) McKay and Fujinaga [2005]	
Feature set: Instrumentation, texture, rhythm, dynamics, pitch statistics, and melody.	
Dataset: 950 MIDI files were collected and hand classified for use in training and testing.	
Classification Algorithm	**Accuracy**
Neural Networks and Support Vector machines (9 classes)	98.00%
Neural Networks and Support Vector Machines (38 classes)	57.00%
(24) McKinney and Breebaart [2003]	

Continued on next page

Table 2 – *Continued from previous page*

Feature set: Two feature sets: Static and Temporal Features. Standard low-level features: RMS, Spectral centroid, bandwidth, zero-crossing rate, spectral roll-off, band energy ratio, delta spectrum, delta spectrum magnitude, pitch, and pitch strength. SLL features (36 of them) were captured in the following frequencies: DC values, 1-2 Hz, 3-15 Hz, 20-43 Hz. 13 MFCCs were captured in the folloiung frequencies: DC values, 1-2 Hz, 3-15 Hz, 20-43 Hz. Psychoacoustic features: average roughness, standard deviation of roughness, average loudness, average sharpness, 1-2 Hz loudness modulation energy, 1-2 sharpness modulation energy, 3-15 Hz loudness modulation energy, 3-15 sharpness modulation energy, 20-43 Hz loudness modulation energy, 20-43 sharpness modulation energy. Audio filter bank temporal envelopes (AFTP): DC envelope values of filters 1-18; 3-15 Hz envelope modulation energy filters 1-18; 20-150 Hz envelope modulkation energy of filters 3-18; 150-1000Hz envelope modulation energy of filters 9-18.	
Dataset: Hand selected popular music from seven different genres: Jazz, Folk, Electronica, R&B, Rock, Reffae, and Vocal. The database used in the current study is a "quntessential" subset of a larger database.	
Classification Algorithm	**Accuracy**
Quadratic discriminant analysis SLL feature set (General audio)	$86 \pm 4\%$
Quadratic discriminant analysis SLL feature set (Genre)	$61 \pm 11\%$
Quadratic discriminant analysis MFCC feature set (General audio)	$92 \pm 3\%$
Quadratic discriminant analysis MFCC feature set (Genre)	$65 \pm 10\%$
Quadratic discriminant analysis AFTE feature set (General audio)	$93 \pm 2\%$
Quadratic discriminant analysis AFTE feature set (Genre)	$74 \pm 9\%$
(25) Meng *et al.* [2007]	
Feature set: Short term features (20-40ms frame size): MFCCs. Midterm features (frame sizes 1000-2000ms): high zero crossing rate, , low short time energy ratio. Long term features (using the full spectrum): beat histogram. Temporal feature integration was conducted using DAR and avoided MeanVar and FC feature integration approaches.	
Dataset: The first data set, denoted "data set A," consists of 100 sound clips distributed evenly among the five music genres: Rock, Classical, Pop, Jazz, and Techno. Each of the 100 sound clips, of length 30 s, are recorded in mono PCM format at a sampling frequency of 22 050 Hz.	
Classification Algorithm	**Accuracy**

Continued on next page

Table 2 – *Continued from previous page*

linear model trained by minimizing least squares error (LM) (Multivariate autoregressive model for feature integration (MAR))	92 ± 1%
LM (Diagonal autoregressive for feature integration (DAR))	89 ± 1%
LM (Filter bank coefficients for feature integration (FC))	85 ± 1%
LM (Mean covariance model for feature integration (MeanCov))	79 ± 1%
LM (Mean variance model for feature integration (MeanVar))	81 ± 1%
Generalized linear model (GLM) with MAR	89 ± 1%
GLM with DAR	88 ± 1%
GLM with FC	85 ± 1%
GLM with MeanCov	86 ± 1%
GLM with MeanVar	89 ± 1%
Gaussian classifier (GC) with MAR	87 ± 1 %
GC with DAR	82 ± 1%
GC with FC	84 ± 1%
GC with MeanCov	86 ± 1%
GC with MeanVar	86 ± 1%
Gaussian mixture model (GMM) with MAR	81 ± 1%
GMM with DAR	83 ± 1 %
GMM with FC	83 ± 1%
GMM with MeanCov	87 ± 1%
GMM with MeanVar	87 ± 1%
(26) Meng *et al.* [2007]	
Feature set: Short term features (20-40ms frame size): MFCCs. midterm features (frame sizes 1000-2000ms): high zero crossing rate, low short time energy ratio. Long term features (using the full spectrum): beat histogram. Temporal feature integration was conducted using DAR and avoided MeanVar and FC feature integration approaches.	

Continued on next page

Table 2 – *Continued from previous page*

Dataset: The second data set denoted "data set B" consists of 1210 music snippets distributed evenly among the 11 music genres: Alternative, Country, Easy Listening, Electronica, Jazz, Latin, Pop&Dance, Rap&HipHop, R&B Soul, Reggae, and Rock. Each of the sound clips, of length 30 s, are encoded in the MPEG1- layer 3 format with a bit-rate of 128 kb/s. The sound clips were converted to mono PCM format with a sampling frequency of 22 050 Hz prior to processing.

Classification Algorithm	Accuracy
LM with MAR	45.5 ± 1%
LM with DAR	38 ± 1%
LM with FC	34 ± 1%
LM with MeanCov	35.5 ± 1%
LM with MeanVar	30± 1%
GLM with MAR	48 ± 1%
GLM with DAR	43.5 ± 1%
GLM with FC	38 ± 1%
GLM with MeanCov	38 ± 1%
GLM with MeanVar	33± 1%
GC with MAR	37 ± 1%
GC with DAR	34 ± 1%
GC with FC	27 ± 1%
GC with MeanCov	27.5± 1%
GC with MeanVar	30± 1%
GMM with MAR	27 ± 1%
GMM with DAR	35.5 ± 1%
GMM with FC	38 ± 1%
GMM with MeanCov	40 ± 1%
GMM with MeanVar	38 ± 1%

(27) Pampalk *et al.* [2005]

Feature set: Spectral similarity, MFCCs, Frame clustering, cluster model similarity

Dataset: ISMIR2004

Classification Algorithm	Accuracy
Spectral similarity described by Pachet and Aucouturier [2004]	82.30%

(28) Panagakis *et al.* [2008]

Feature set: Multiscale spectro-temporal modulation features

Dataset: ISMIR2004

Classification Algorithm	Accuracy

Continued on next page

Table 2 – *Continued from previous page*

Non-Negative Tensor Factorization	80.47%
High Order Singular Value Decomposition	80.95%
Multilinear Principal Component Analysis	78.53%

(29) West and Cox [2004]

Feature set: Mel-frequency cepstral coefficients, octave-scale spectral contrast feature, Marsyas-0.1 single vector genre feature set, reducing covariance in calculated features, modelling temporal variation

Dataset: In this evaluation, we have used six classes of audio, each represented by 150 samples, which were a 30 second segment chosen at random from a song, also chosen at random from a database composed of audio identified by the authors as being from that genre of music. The first 10 seconds of each piece is ignored as this sometimes contains little data for classification. The genres selected were Genres: Rock, Classical, Heavy Metal, Drum and Bass, Reggae and Jungle music. 50% training 50% testing.

Classification Algorithm	Accuracy
Single Gaussian Model	63 ± 1%
Gaussian Mixture Model	62 ± 1%
Fisher Criterion Linear discriminant Analysis	45.5 ± 1%
Classification trees with Linear discriminant Analysis	67.5 ± 1%
Classification trees with single Gaussians and Mahalanobis distance measurements	67 ± 1%

(30) Xu *et al.* [2003]

Feature set: Beat Spectrum, LPC derived cepstrum, zero-crossing rate, spectrum power, Mel-frequency cepstral coefficients.

Dataset: The music dataset used in musical genre classification experiment contains 100 music samples. They are collected from music CDs and Internet and cover different genres such as classic, jazz, pop and rock. All data are 48.0kHz sample rate, stereo channels and 16 bits per sample. In order to make training results statistically significant, training data should be sufficient and cover various genres of music.

Classification Algorithm	Accuracy
Support Vector Machines	93.64%

(31) Xu *et al.* [2005a]

Feature set: Mel-frequency cepstral coefficients, spectrum flux, cepstrum flux, spectral power, amplitude envelop, linear prediction coefficients (LPC), (LPC)-derived cepstrum coefficients (LPCC), zero-crossing Rates.

Continued on next page

Table 2 – *Continued from previous page*

Dataset: "They are collected from music CDs and the Internet and cover different genres such as pop, classical, rock and jazz. Test Set 1: The first part contains 20 pure music samples (40 000 frames)"	
Classification Algorithm	**Accuracy**
Support Vector Machines	99.83%
Hidden Markov Model	96.44%
Nearest Neighbours	80.34%
(32) Xu *et al.* [2005a]	
Feature set: Mel-frequency cepstral coefficients, spectrum flux, cepstrum flux, spectral power, amplitude envelop, linear prediction coefficients (LPC), (LPC)-derived cepstrum coefficients (LPCC), zero-crossing rates.	
Dataset: "They are collected from music CDs and the Internet and cover different genres such as pop, classical, rock and jazz. Test Set 2: The second part contains 20 vocal music samples (40 000 frames)."	
Classification Algorithm	**Accuracy**
Support Vector Machines	93.34%
Hidden Markov Model	92.77%
Nearest Neighbours	77.67%
(33) Xu *et al.* [2005a]	
Feature set: Mel-frequency cepstral coefficients, spectrum flux, cepstrum flux, spectral power, amplitude envelop, linear prediction coefficients (LPC), (LPC)-derived cepstrum coefficients (LPCC), zero-crossing rates.	
Dataset: They are collected from music CDs and the Internet and cover different genres such as pop, classical, rock and jazz. Test Set 3: The third part contains 15 pure music samples and 10 vocal music samples (50 000 frames).	
Classification Algorithm	**Accuracy**
Support Vector Machines	96.02%
Hidden Markov Model	89.13%
Nearest Neighbours	78.78%

3.2 CONTENT-BASED FEATURE EXTRACTION

In this section the major contributions of content-based acoustic feature extraction from audio are briefly reviewed. Recall that content-based acoustic features can be classified as timbre, rhythmic, and pitch features [Tzanetakis and Cook 2002], and so the following related work - in this section - employ a mixture of these content features for music genre classification.

Foote [1997], probably one of the first authors in content based audio retrieval, studied similarities between musical content/audio and used these similarities to generate sound *queries*. By doing so, Foote [1997] presented a search engine which was used to retrieve music content from a database based on these similarities. Foote [1997] proposed two similarity functions as a composition of distance measures. Another early author in music style recognition, Dannenberg *et al.* [1997], studied machine learning algorithms for music content style classification. Dannenberg *et al.* [1997] hypothesised that a machine can perform extraction techniques to learn metrics from an audio signal and use these metrics, along with machine learning algorithms, to build musical style classifiers. Li [2000] also used distance metrics for music information retrieval by proposing a *nearest feature line method* for content-based genre classification. Liu and Huang [2000], however, produced a more sophisticated method for context based indexing by proposing a novel metric for distance between two Gaussian mixture models. Logan and Salomon [2001] used the K-means clustering algorithm on Mel-frequency cepstral coefficients features along with another novel comparison metric for content-based audio information retrieval. Pampalk *et al.* [2003] conducted a brief comparison between long-term and short-term feature descriptors on various datasets, including those novel features belonging to Logan and Salomon [2001] and Aucouturier and Pachet [2002]. The results from Pampalk *et al.* [2003] informs this research as analysing spectral histograms thorough large scale evaluation have been proven most effective. Zhang and Kuo [2001] classified audio signals from popular TV series and movies by using a heuristic rule-based system that involved content-based features.

The next three sections further analyse the contributions towards content-based feature extraction by examining the development of timbre, rhythmic, and pitch content-based features respectively.

3.2.1 *Timbre Content-based Features*

Timbre content-based features has its roots in traditional speech recognition as it is considered a key descriptor in speech classification. Customarily, timbre descriptors are obtained by first performing a short-time Fourier transform (STFT) and then extracting from every short-time frame or window a descriptor [Rabiner and Juang 1993]. Timbre features include but are not limited to spectral centroid, rolloff, flux, energy, zero crossing rate, and Mel-Frequency Cepstral Coefficients (MFCCs) [Rabiner and Juang 1993]. The most effectual timbre feature used in speech recognition are the MFCCs extracted form the spectral histogram. Logan [2000] examined these MFCCs to model musical content for music/speech classification. Based only on timbre features, Deshpande *et al.* [2001] classified music datasets into rock, piano, and jazz using GMMs, SVMs, and the k-NN algorithms. Foote [1997] performed audio retrieval using simple audio features, including energy, along with 12 MFCCs by constructing a learning tree vector quantizer. Aucouturier and Pachet [2002] planned to introduce a timbral similarity measure only for MFCCs to classify various sized music databases using GMMs. Unfortunately, the timbral similarity measure was not applicable to large datasets and so Aucouturier and Pachet [2002] proposed a measure of "interestingness" for MFCC features. Li *et al.* [2003] proposed a novel feature

set using Daubechies Wavelet Coefficient Histograms which proved to be suitable to categorise amplitude variations in musical signals for music genre classification.

3.2.2 *Rhythmic Content-based Features*

Often when one wants to examine musical content an obvious considerable factor is rhythm. Rhythmic content-based features refer to the regularity of the rhythm, beat and tempo of the audio signal. Various authors have successfully explored rhythmic features in music genre classification literature [Goto and Muraoka 1994; Laroche 2001; Scheirer 1998]. Foote and Uchihashi [2001] represented rhythm by using a beat spectrum. Lambrou *et al.* [1998] classified music datasets into rock, piano, and jazz by using temporal domain statistical features along with various wavelet domain transforms. Soltau *et al.* [1998] showed that temporal abstract features can be learnt by neural networks as representing temporal structures of an input signal, which could be used for music genre identification. A question often arises whether or not there exists a difference between music and speech discrimination. Saunders [1996]; Scheirer and Slaney [1997] present a clear discrimination between music and speech identification. Dixon *et al.* [2004] experimented with rhythmic patterns combined with additional features derived from them. Meng *et al.* [2007] developed a multivariate autoregressive model to model temporal feature correlation.

3.2.3 *Pitch Content-based Features*

Pitch content-based features describe the frequency statistics associated with (musical) signal bands that are sampled using pitch detection or extraction procedures. Pitch content-based features can be expressed as absolute pitch or interval pitch [McNab *et al.* 1996]. Both of these expressions can be quantified as pitch content-based acoustic features [Uitdenbogerd and Zobel 1999].

The duration and differences in time onsets for two consecutive notes are expressed as IOIR and IOI respectively. Pardo and Birmingham [2002] used the logarithm of variations of IOIR and IOI to perform music identification. McNab *et al.* [1996]; Uitdenbogerd and Zobel [1999] have shown that using interval pitch and IOIR on different melodies[5] can yield transpositions and time invariance that take advantage of both pitch and duration to identify melody sequences as more unique. Kotsifakos *et al.* [2012] conducted an intense survey on Query-By-Humming (QBH) similarity techniques based on pitch content-based features. Adams *et al.* [2004] used absolute pitches with Dynamic Time-Warping (DTW) for whole sequence matching. The problem with whole sequence matching is that of tempo variations, which was solved by scaling the target sequences before applying the DTW [Mongeau and Sankoff 1990; Mazzoni and Dannenberg 2001]. Lemström and Ukkonen [2000] used dynamic programming to embed transposition invariance as a cost function, however, it is often

5 or variations of these descriptors.

the objective to perform whole query matching without using duration. Bergroth *et al.* [2000]; Uitdenbogerd and Zobel [1999] provided an alternative dynamic programming approach called the Longest Common SubSequence (LCSS)[6] to asses melodic similarity. Iliopoulos and Kurokawa [2002] provided a comprehensive algorithm using absolute pitches for whole query matching for music identification. The algorithm presented by Iliopoulos and Kurokawa [2002] allowed for a constant restricted amount of gaps and mismatches.

Zhu and Shasha [2003] provided an unusual approach to whole sequence matching by dividing a song in a dataset into chunks where a query is resolved by comparing it to each chunk. Following the work by Zhu and Shasha [2003], Hu *et al.* [2002]; Jang and Gao [2000] addressed whole sequence matching by only using absolute pitches and tempo scaling upon these chunks. Edit distances (ED) with its respective variations have been proven to be a useful tool for music retrieval [Lemström and Ukkonen 2000; Pauws 2002]. Kotsifakos *et al.* [2011]; Unal *et al.* [2008] use ED, taking advantage of pitch and duration statistics, to perform QBH. More recent methods to resolve QBH uses SPRING [Sakurai *et al.* 2007] to find chunks from evolving numerical streams to partly match the query [Kotsifakos *et al.* 2011].

3.3 RELATED CLASSIFICATION TECHNIQUES

There have been several attempts to conduct automatic music genre classification using the following supervised classification techniques: K-nearest neighbours (KNN) [Tzanetakis and Cook 2002; Li and Ogihara 2006; Haggblade *et al.* 2011], Gaussian Mixture Models (GMM) [Tzanetakis and Cook 2002; Li and Ogihara 2006; West and Cox 2004], Linear Discriminant Analysis (LDA) [Li and Ogihara 2006], Adaboost [Bergstra *et al.* 2006], Hidden Markov Models (HMM) [Kim *et al.* 2004], Regularised Least-squares Framework [Song and Zhang 2008], and Support Vector Machines (SVM) [Li and Ogihara 2006; Panagakis *et al.* 2008; Xu *et al.* 2005a]. Although these techniques obtain good accuracy and present thorough methodologies, it is hypothesised that many of them can be improved from feature extraction to automatic classification.

Several classification techniques have been explored to perform music retrieval using content-based acoustic feature extraction [Tzanetakis and Cook 2002; Meek and Birmingham 2004; Soltau *et al.* 1998; Basili *et al.* 2004; Hamel *et al.* 2011]. Tzanetakis and Cook [2002] used K-nearest neighbours and Gaussian mixture models for music genre classification based on a novel set of content-based features. Following attempts to perform classification for QBH, Meek and Birmingham [2004]; Shifrin *et al.* [2002]; Unal *et al.* [2008] presented a HMM approach.

Soltau *et al.* [1998] extended the work of Meek and Birmingham [2004]; Shifrin *et al.* [2002]; Unal *et al.* [2008] for music genre classification maintaining the HMM

6 This approach allows gaps on both sequences during alignment.

approach. While HMMs have been proven useful for music composition[7] [Khadkevich and Omologo 2009; Pearce and Wiggins 2007], it is unlikely that HMMs will be suitable to model classification of musical genre simply because the training is computationally expensive given the dimensionality of the problem[8] and size of the dataset[9].

Berenzweig *et al.* [2003] presented a model that mapped the genre and composer of a corpus constructed dataset to a Cartesian plane using MFCCs as a feature descriptor. The design matrix was represented using the posterior probabilities of different neural networks in multi-dimensional space. This representation was then used to classify the corpus construction using Gaussian mixture models and KL-divergence algorithms were used for comparisons between melodies. Berenzweig *et al.* [2003] produced a comprehensive algorithm, however, the accuracy obtained by this methodology was unsatisfactory.

McKay and Fujinaga [2005] used a feature significance method where features were leveraged or ranked by their important characteristics. These ranking or weight systems were used to appropriately weigh the features in final calculations. McKay and Fujinaga [2005] heavily leveraged content-based features for music genre classification (using k-NN, neural networks and combination of several classifiers) which lost the significance of the weighing system as the application seemed to not need the other features given their insignificant weighing. The use of priority features by McKay and Fujinaga [2005] informs this research, however, the weighing system in final calculations will not be done as this research hypothesises that it is better to pre-process feature significance (ranking system) and by only using the features with significant leverage. Therefore this research will consider adopting wrapper and filter feature selection strategies.

Cataltepe *et al.* [2007] transformed melodies from MIDI formats to alphabetic-strings and audio signals. The alphabetic strings employed a normalised compression distance equation to differentiate between melodies, while the content-based features were extracted from the audio signal. Cataltepe *et al.* [2007] used LDC and k-NN classifiers, in some cases mixed and independently, with appropriate weights to perform the final genre classification. Multi-class support vector machine classifiers are very popularly used in genre classification [Lidy and Rauber 2005; Mandel and Ellis 2005; Hamel *et al.* 2011] along with many other interesting approaches [Bergstra *et al.* 2006; Cilibrasi *et al.* 2004; Dannenberg *et al.* 1997; Tzanetakis and Cook 2002]. David [2000] explored Tree-based Vector Quantisation (VQ) and GMMs [Turnbull *et al.* 2007] for music genre classification. Following David [2000], McFee *et al.* [2012] explored k-NN on frame-level MFCC features and used the cluster centers for VQ. Berenzweig *et al.* [2004] presented an extensive study on subjective and content-based features. Their study compared several similarity metrics on orthodox datasets. Basili *et al.* [2004] studied several classification algorithms on different datasets for music genre classification.

7 As this is nothing more than observing probabilistic behaviour which HMMs are most suited for.
8 Particularly for large music datasets that we will have to deal with, e.g. MusicBrainz with 16 000 000 entries.
9 The number of music tracks and the length of the sequences used to train the model

3.4 MUSIC INFORMATION RETRIEVAL

Due to the extensive research to improve music genre classification for music information retrieval and recommendation, there has been a demand for common evaluation to assess and compare different author's contributions [Downie 2003]. As a result of this, many music information retrieval (MIR) contests have emerged. The first of which was an ISMIR contest held in 2004 to evaluate five different components of MIR research - genre classification being one of them. In 2005 ISMIR continued as the MIREX contest.

3.5 CONTRIBUTIONS

Music genre classification is a rudimentary part of MIR and music recommendation systems. Thorough the availability of digital music on the Internet and significant customer and industry related contributions, it has been an expanding field of research [Li *et al.* 2003]. For many years music genre classification has been done by hypothesising features extracted from an audio signal to be true descriptors of music. These features are obtained by listening to an audio signal and then attempting to replicate the auditory system though a computational model. Many authors have discovered truly useful features but because of the overflow of ideas, these features have never been quantitatively compared and analysed for their individual importance. Furthermore, it is often the case where the discovery of features useful in some fields[10], can be unknowingly useful in other fields.

Example 3.2. For example, *energy* - which is a prevalent speech processing feature - might me unknowingly useful in music genre classification, or Mel Frequency Ceptral Coefficients (MFCCs) might not be as useful as researchers might have hypothesised.

Therefore, this research plans to contribute to current knowledge in music genre classification by ranking features so future researchers will know which features are more effective than others using a benchmark dataset (GTZAN).

This research will use six classification techniques, two of which have never been used for genre classification: random forests and linear logistic regression models. Linear logistic regression models provides the best classification for genre detection on GTZAN genres. According to an extensive literature review, only three other authors provide better classification using different techniques with the same BOF setting [Sturm 2013b; Bergstra *et al.* 2006; McKay and Fujinaga 2005]. Briefly stated, the proposed research aims to provide the following contributions:

1. A list of existing methods for genre classification, these include more than 42 noteworthy content-based classification methods that are compared in terms of their feature-sets, dataset, and classification techniques.

10 For example, speech recognition or audio compression

2. An extensive collection of features with two classification algorithms never used for genre classification. Table 3 shows a list of some features and classification algorithms used by previous authors for music genre classification. The first column is the feature or classification algorithm and the second column is the author who used the corresponding feature. In the second column a ✯ indicates that the feature or classification algorithm was not used for genre detection before. This research will attempt to use these features/classifiers for genre detection.

3. A thorough comparison of feature representation methods as well as a suitable representation for each presented genre detecting feature will be provided. These feature representations include the arithmetic mean; MFCC; 3, 5, 10, 20, 30 -bin feature histograms; and area methods (better explained in Chapter 5).

4. A complete design matrix with all parameters that achieve 81% classification using linear regressions on 10 GTZAN genres, along with the corresponding confusion matrix. This research also emphasises which features are most suited to classify each GTZAN genre.

Table 3: List of features and classifiers used by previous authors.

Attempting New Classifiers and Features for Genre Classification	
Content-based features	
Amplitude modulation	Lidy and Rauber [2005]; Mandel and Ellis [2007]; McKinney and Breebaart [2003]; Panagakis *et al.* [2008]
Energy	Lidy and Rauber [2005]; Gouyon *et al.* [2000]; Hamel *et al.* [2011]; Lee *et al.* [2007]; McKinney and Breebaart [2003]; Li *et al.* [2003]; Meng *et al.* [2007]; Tzanetakis and Cook [2002]; Sturm [2013b]
Autocorrelation coefficients	Benetos and Kotropoulos [2008]; Lambrou *et al.* [1998]
Mel-frequencies Cepstrum coefficients	Mandel and Ellis [2007]; McKinney and Breebaart [2003]; Benetos and Kotropoulos [2008]; Berenzweig *et al.* [2003]; Lee *et al.* [2007]; Li *et al.* [2003]; Meng *et al.* [2007]; Tzanetakis and Cook [2002]; Berenzweig *et al.* [2003]; Cast *et al.* [2014]; Guo and Li [2003]; Holzapfel and Stylianou [2008]; Mandel and Ellis [2005]; Meng *et al.* [2007]; West and Cox [2004]; Xu *et al.* [2005a]; Sturm [2013b]
Spectral decrease	✯
Spectral flatness	Benetos and Kotropoulos [2008]

Spectral flux	Li *et al.* [2003]; Tzanetakis and Cook [2002]; Xu *et al.* [2005a]; Sturm [2013b]
Spectral variation	✯
Temporal centroid	Benetos and Kotropoulos [2008]
Temporal spread, skewness, and kurtosis	✯
Zero-crossing rate	Bergstra *et al.* [2006]; Gouyon *et al.* [2000]; Li *et al.* [2003]; McKinney and Breebaart [2003]; Meng *et al.* [2007]; Tzanetakis and Cook [2002]; Xu *et al.* [2005a]; Sturm [2013b]
Spectral derivative	✯
Complex domain onset detection	✯
Linear predictor coefficients	Xu *et al.* [2005a]
Line spectral frequency	✯
Loudness	Benetos and Kotropoulos [2008]; Lidy and Rauber [2005]; McKinney and Breebaart [2003]
Compute octave band signal intensity (with ratio)	✯
Perceptual sharpness and spread	✯
Spectral crest factor (per band) + peak-based features	✯
Spectral flatness per band	✯
Spectral rolloff	Benetos and Kotropoulos [2008]; Bergstra *et al.* [2006]; Li *et al.* [2003]; McKinney and Breebaart [2003]; Tzanetakis and Cook [2002]; Sturm [2013b]
Envelope centroid	Mandel and Ellis [2007]
Envelope spread, skewness, kurtosis	✯
Spectral centroid	Benetos and Kotropoulos [2008]; Bergstra *et al.* [2006]; Li *et al.* [2003]; McKinney and Breebaart [2003]; Tzanetakis and Cook [2002]; Sturm [2013b]
Spectral spread, skewness, kurtosis	Benetos and Kotropoulos [2008]; Bergstra *et al.* [2006]; Lambrou *et al.* [1998]
Spectral slope	✯
Compactness	✯
Fraction of low energy	✯

Beat histogram features	Tzanetakis and Cook [2002]; Cataltepe *et al.* [2007]; Li *et al.* [2003]; Meng *et al.* [2007?]; Xu *et al.* [2003]
Chroma	☆
Relative difference function	☆
Dataset	
GTZAN Genre Collection	Benetos and Kotropoulos [2008]; Bergstra *et al.* [2006]; Cast *et al.* [2014]; Holzapfel and Stylianou [2008]; Li *et al.* [2001 2003]; Lidy *et al.* [2007]; Panagakis *et al.* [2008]; Tzanetakis and Cook [2002]
Classification Algorithms	
Naïve Bayes [John and Langley 1995]	Basili *et al.* [2004]; Cast *et al.* [2014]
Support vector machines [Chang and Lin 2001; EL-Manzalawy 2005]	Benetos and Kotropoulos [2008]; Deshpande *et al.* [2001]; Guo and Li [2003]; Lidy and Rauber [2005]; Mandel and Ellis [2005 2007]; McKay and Fujinaga [2005]; Xu *et al.* [2005a]
Gaussian mixture models	Berenzweig *et al.* [2003]; Deshpande *et al.* [2001]; Holzapfel and Stylianou [2008]; Meng *et al.* [2007]; Tzanetakis and Cook [2002]; West and Cox [2004]
Multilayer perceptron	Benetos and Kotropoulos [2008]
Linear logistic regression models [Sumner *et al.* 2005; Landwehr *et al.* 2005]	☆
K-nearest neighbours [Aha and Kibler 1991]	Basili *et al.* [2004]; Cast *et al.* [2014]; Deshpande *et al.* [2001]; Guo and Li [2003]; Lee *et al.* [2007]; Tzanetakis and Cook [2002]; Xu *et al.* [2005a]
JRIP [Cohen 1995]	Basili *et al.* [2004]
J48 [Quinlan 1993]	Basili *et al.* [2004]
Forest of random trees [Breiman 2001]	☆

All of the required tools for the proposed methodology are now defined. Chapter 4 shows how a new methodology can be used for genre classification by using a novel feature selection and various machine learning techniques for supervised classification.

4

THE RESEARCH DESIGN

4.1 RESEARCH DESIGN

This *Research* is divided into two main parts: *the feature extraction step* and *the classification step*. Figure 2 shows an overview of the research design. The feature extraction step involves an audio dataset (e.g. GTZAN) which contains all of the audio data to be included by the experiment. A list of defined features are used to extract descriptions from every audio file in the training dataset. The features and dataset together give us the design matrix with dimensions $n \times (m+1)$[1]. Each row of the design matrix represents one and only one audio file description and each column of the design matrix represents a feature value. A feature value is a value obtained after a feature extraction. The design matrix also contains the genre label for every audio signal, once the design matrix is complete, it is then passed to the *classification step*.

Part II of the research design is called the *classification step*. In this step various classification algorithms will be used to classify the labels from the design matrix based on the feature values. As seen in Figure 2 there will be more than one classification technique applied to the design matrix. The set of classification algorithms used by this research include:

1. **Support Vector Machines** (SVM)
2. **Linear Logistic Regression Models** (LLRM)
3. **K-Nearest Neighbours** (KNN)
4. **Naïve Bayes** (NB)
5. **Random Forests** (RF)
6. **Multilayer Perceptrons** (MP)

A detailed description of these classification algorithms are given in the Appendix C of this dissertation. Lastly, a comparison of strategies is done so the optimal classification technique(s) will be declared along with the selected features. The following factors will guide the analysis of the classification algorithms used:

1 Where n is the number of audio files, m is the number of features, and the last column is the genre label.

1. **Confusion Matrices:** A confusion matrix, commonly referred to as a contingency matrix, is a tool for visualisation between the predicted class and the actually classified class. This allows us to see clearly how some genres have been classified as others, which helps us analyse the performance of a learning model.

2. **Success Rates:** The classification accuracy or success rate tells us what percentage of pieces were correctly classified.

The next two sections, Section 4.1.1 and Section 4.1.2, describe the two main components illustrated in Figure 2, in terms of Part I and Part II respectively; feature extraction techniques are explored in Section 4.1.1; Section 4.1.2 extends current binary classification techniques mentioned above for multi-class classification; and finally, the dataset used for this analysis will be presented in Section 4.1.3.

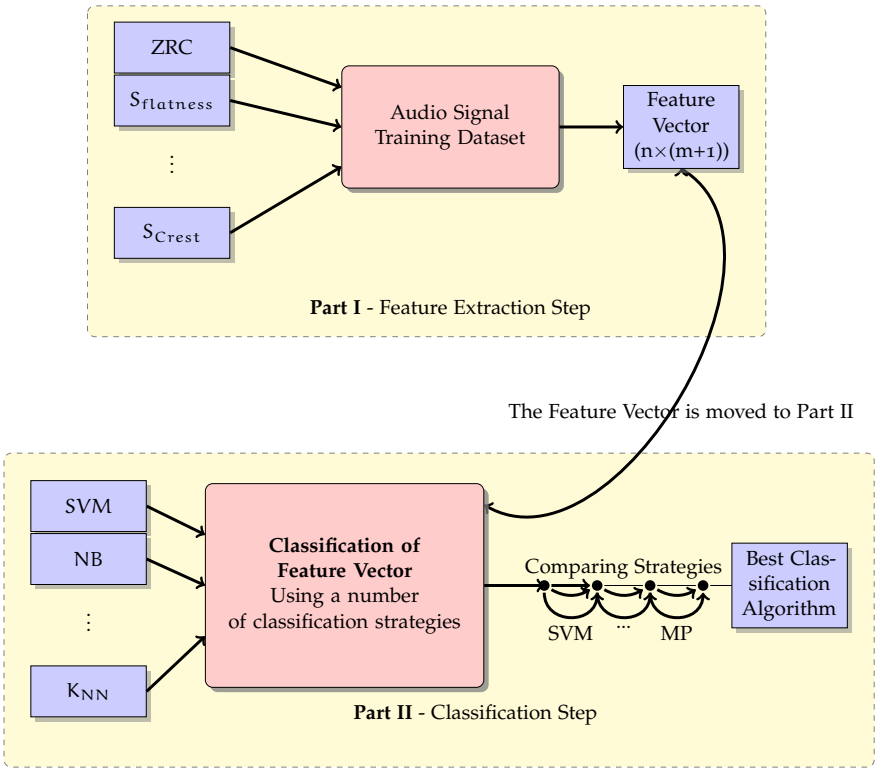


Figure 2: Research design overview

4.1.1 *Features*

In order to achieve genre classification from a dataset of discrete-time signals, feature extraction must be performed. Feature extraction uses measurements to obtain information from signal data. This information describes the signal and is used to classify the elements of the dataset. A list of spectral features used by this research is given by Table 4.

Table 4: A general list of features.

Proposed Features
Spectral flux, variability, decrease, flattness, slope, centroid, rolloff, and variation
Shape, envelope, and temporal statistics (centroid, kurtosis, skewness and spread)
Compactness
Mel-frequency Cepstral coefficients
Peak centroid, flux, smoothness, and crest factor
Complex domain onset detection
Loudness (+ sharpness and spread)
OBSI (+ Ratio)
Autocorrelation coefficients
Amplitude modulation
Zero crossing (with the strongest frequency of zero crossings)
Linear predictor coefficients and Line spectral frequency
Energy
Chroma

A less detailed overview of the feature extraction process is given by Figure 3. The upper layer represents the discrete-time signal as a hyperplane. Each square in this hyperplane represents a numeric value which together make up the entire signal. In the lower layer the smaller blocks are made from the larger ones by applying a signal transform to the DTS. Such signal transforms include the fast Fourier transform, constant-Q or even energy. These signal transforms serve as suitable interpretations of the signal and are used to create families of features.

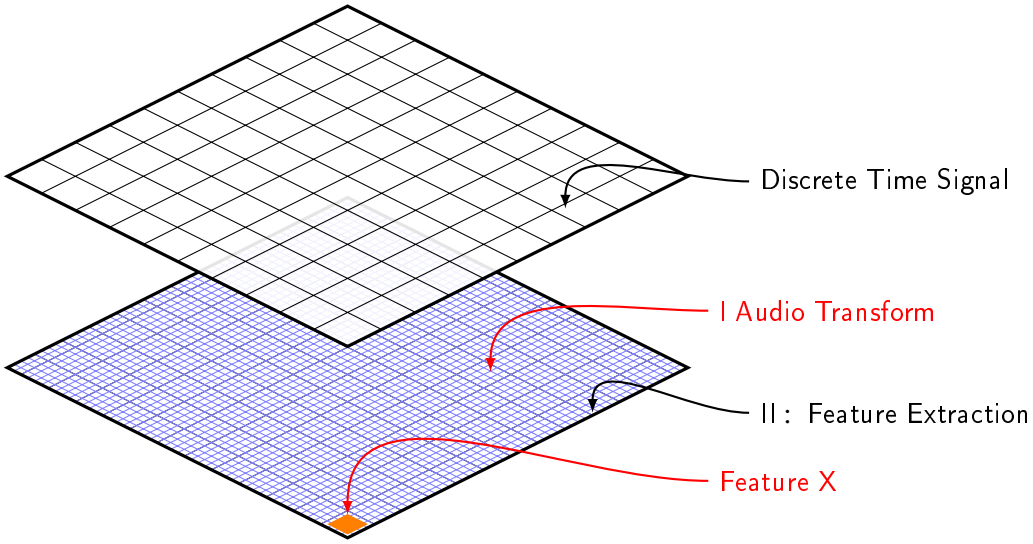


Figure 3: Basic feature extraction process.

The large block (orange) in the second layer indicates a feature being derived from the signal transform, letâs call this feature X. Using only a few blocks of the signal transform we were able to create feature X, and so from this signal transform we can create many instances of the same feature X. Therefore we could land up with 10 000 values for feature X, and so we need some way to represent this feature more compactly. In Chapter 5 we explore different ways to do this and apply many representations for features as many features are best represented differently. Chapter 6, Chapter 7, Chapter 8, and Chapter 9 introduce different signal transforms and their respective feature families derived from them.

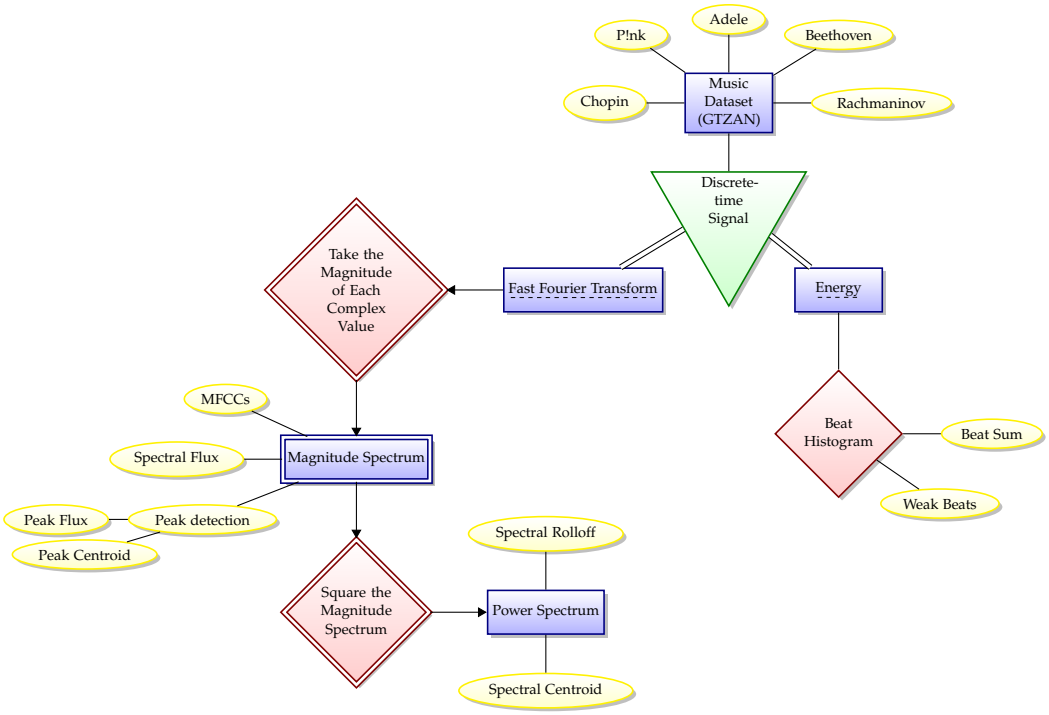


Figure 4: More detailed feature extraction process.

Figure 4 shows a more detailed process of feature extraction. The uppermost box resembles a music dataset, such as GTZAN, this dataset contains a collection of music signals each with a known genre. For every music signal in the dataset all the signals windows are examined by a signal transform, such as FFT or Energy. These signal transforms yield new spectra that given us different features which are added to the design matrix. Often simple modifications can yield new powerful spectra. For example, in the case of Figure 4, the magnitude spectrum was squared to yield to power spectra, which give us spectral rolloff and centroid.

4.1.2 *Multi-class Classification*

Binary classification techniques need to be extended to accommodate multiple classes if one is to classify more than two genre types. A way to achieve this classification is to simply consider multiple *binary* classification problems as opposed classifying one complex multi-class dataset [Duan and Keerthi 2005]. Therefore, some mapping must be achieved in order to model this reduction and extraction of binary classification

problems from the multi-class set. Duan and Keerthi [2005] and Hsu and Lin [2002] proposed such reduction techniques. These techniques involve building one of two binary classification problems: *one-verses-all* or *one-verses-one*.

4.1.2.1 *The One-verses-all Paradigm*

In the *one-verses-all* paradigm, one class is selected as class 0 and every other class is modeled as class 1. Classes 0 and 1 are then classified using the binary classification technique proposed. Further work should be done to model every other class in this way. The classifications are merged to form what appears to be a multi-class classification. If new data points are introduced to the multi-class classification model, a *winner-takes-all strategy* is employed where the classifier with the highest output function declares where the point will be classified.

4.1.2.2 *The One-verses-one Paradigm*

In the *one-verses-one* paradigm, each class is modelled with respect to every other class. The classification is then done by using the binary classification technique mentioned above. The classifications are again merged to form what appears to be a multi-class classification. If new data points are introduced to the multi-class classification model, a *max-wins voting strategy* is employed where every classifier allocates a possible class label to each respective class. Finally, a voting strategy is used to count the classifier allocations and the class label with the highest vote-count decides the ultimate classification.

Further attempts to convert multi-class classification to binary classification problems by Platt *et al.* [1999] and Dietterich and Bakiri [1995] include Directed Acyclic Graph SVM (DAGSVM) and item error-correcting output codes respectively. Instead of decomposing multi-class classification problems into binary classification problems, Lee *et al.* [2001 2004]; Crammer and Singer [2002] proposed a multi-class SVM method which using optimization algorithms to solve multi-class classification problems.

4.1.3 *GTZAN Dataset*

In a recent literature review of music genre recognition (MGR) by Sturm [2012b], it was seen that in the 467 published works, most of MGR experiments involved used the GTZAN dataset. GTZAN is a compilation of 1000 30-second music excerpts clearly categorized into 10 labelled genres. Although the GTZAN dataset has been used very frequently, a recent study by Sturm [2012a] has showed that the dataset contains several faults particularly in repetitions, mislabelling, and distortions. According to Sturm [2013a], these faults challenge the interpret-ability of any result derived by using GTZAN. Sturm [2013a] maintains that these faults do affect all MGR systems in the someway and that performances related to GTZAN are still meaningfully comparable to MGR systems since they all contain the same faults. Some authors who have used the GTZAN dataset include: Benetos and Kotropoulos [2008];

Bergstra *et al.* [2006]; Cast *et al.* [2014]; Holzapfel and Stylianou [2008]; Li *et al.* [2001 2003]; Lidy *et al.* [2007]; Panagakis *et al.* [2008]; Tzanetakis and Cook [2002]. This is the dataset used in this work.

4.1.4 *Feature Selection*

In order to select the most effective features from a design matrix, some feature selection methods must be explored. Two very common methodologies are the *wrapper* and *filter methods* to select the most effective features for a given dataset and classification algorithm.

4.1.4.1 *The Wrapper Method*

The wrapper method uses a subset evaluator that creates a set of subsets from the design matrix, thereafter, uses a classification algorithm (e.g. support vector machines) to induce classifiers from the features in each subset. It then considers the subset of features with which the classification algorithm runs the best. For example if we have 12 features - the subset evaluator will try to find every possible subset from those 12 features, perhaps it produces three subsets the first one contains 3 features, the second one contains 5 features, and the third one contains seven features. The first subset of features will be applied on the training set with a classification algorithm which will deduce the classification accuracy. On the bases of the three feature subsets, the wrapper algorithm will present the best subset with the best classification accuracy. The selection of the feature subset solely depends on the search technique applied. Some search techniques include random search, depth first search, breath first search, or a hybrid search which integrates the two of them.

4.1.4.2 *The Filter Method*

Unlike the wrapper method the filter method uses a feature evaluator, which evaluates the attributes of features, and a ranking algorithm to rank all the features in a dataset. The ranking algorithm assigns a numeric ranking to each feature in association with an attribute evaluator. After ranking the features one can omit them on a one-at-a-time bases to evaluate the predictive accuracy of the classification algorithms. An issue with the filter method is that the weights put by the ranking algorithm can be very different compared to the weights put by the classification algorithm. There is a danger of over-fitting as the weights provided by the ranking algorithm may not always match the relative importance of the features in the classifier. The ranking algorithm is used to rank features and by omitting one feature at a time from the rank-list one can see how the classification algorithms are performing on the dataset with the features that are not omitted.

Part II

FEATURE ANALYSIS

Wold *et al.* [1996] describes acoustic content as comprising of instrument sounds, speech sound, and environmental sounds. When a human listener wants to classify a piece of music by genre, she will try to identify characteristics (instrument, speech, environment [Wold *et al.* 1996]) in the music, and see if these characteristics/features are similar to those belonging to her previous experience of music from the same genre. Similarly, for a computer to correctly classify a piece of music by genre, the computer must try and identify and compare characteristics/features that exist within a piece of music to those previous classified. Unfortunately, a computer cannot independently or naturally identify features as good or bad to create feature extraction mechanisms. Consequently, these feature extraction mechanisms must be given to the computer to use.

This part presents several features that are hypothesised to be characteristics that can be used to correctly classify musical genre. This part further organises music genre discriminating features into four main components: *The Magnitude Spectrum*, Chapter 6, where timbrel features that describe loudness, noisiness, compactness, e.t.c. are presented; *Tempo Detection*, Chapter 7, where methods that explore rhythmic aspects of the signal are provided; *Pitch Detection*, Chapter 8, where algorithms that describe the pitch of music signals are presented; and finally *Chordal Progressions*, Chapter 9, where we explore chroma as a chordal (environmental) distinguishing feature.

5

FEATURE REPRESENTATION

In contrast to feature selection, which reduces the number of features used for classification, feature representation can be used to approximate a high dimensional design matrix into a lower dimensional one by combining the content of each feature in such a way to preserve the intended feature description. These lower dimensional design matrices are referred to as simplified representations of the original features. A very simple representation is feature averaging, where an n dimensional design matrix will be represented as a 1-dimensional one using an expected value to measure the central tenancy characterized by the higher dimensional design matrix's probability distribution[1]. Other feature representations we explore in this dissertation include MFCC representations, feature histograms, and area moments.

5.1 INTRODUCTION

In this chapter we review feature averaging, histogram strategies and introduce effective method for feature representation using Mel-frequency cepstral coefficients (MFCCs), although feature representation strategies are usually optimized based on the individual feature's distribution. If one assumes that an audio signal remains stationary[2], then some expression can be used to measure the discrete-time signal's (DTS) local characteristics.

Remark 5.1. These measurements can be used to describe the *timbre* characteristics of a DTS. Timbre refers to the character or quality of a musical sound or voice as distinct from its pitch and intensity [Oxford 1989]. Timbre measurements can be used to describe different qualities of the DTS (e.g. intervals, speech etc.) depending on the defined expression.

These measurements, however, contain a large array of values as they are calculated upon each window, frame, or envelope of the DTS. Therefore, some representation needs to be established to approximate the feature description (e.g. timbre measurement) to a compact and small set of values (10, 4, or even 1 value). Before we can explore different feature representation strategies a DTS is firstly defined.

1 Example of basic representations include the arithmetic mean, median, mode, geometric mean, weighted mean e.t.c.

2 Signal's statistical properties do not differ with time.

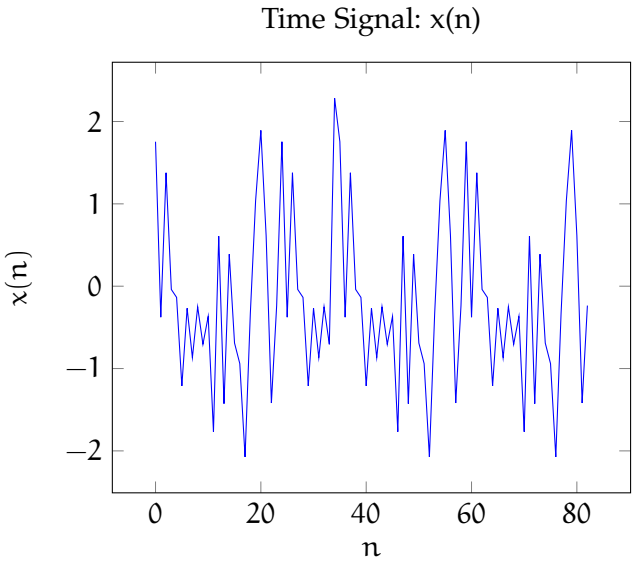


Figure 5: A discrete-time signal.

Definition 5.2. *Discrete-Time Signal*
A real discrete-time signal (DTS) is defined as any time-ordered sequence of real numbers. A real discrete-time signal can be expressed as a real function, $x : \mathbb{Z}^+ \to \mathbb{R}$, defined as:

$$x(n), n \in \mathbb{Z}^+, \tag{1}$$

where $x(n)$ is the n^{th} real number of the signal, and n represents time as a positive integer. Figure 5 shows an example of a discrete-time signal.

Using 3 timbre features on GTZAN genres, the centroid, rolloff and energy in each window were extracted. Figure 6 shows the case count for each extracted design matrix. In Figure 6 there are 938 480 values for each feature extracted from every window in the DTS, because there are so many feature values in this set we need a way to reduce the sample size to a more manageable one. Using the random sampling method to select 50 000 cases we obtain Figure 7. Figure 35 shows a description of the reduced sample size and Figure 36 shows the feature histograms of spectral centroid and rolloff respectively in the appendix of this dissertation[3].

5.2 TEST FOR NORMALITY

In order to use the arithmetic mean as a feature representation effectively it is important that the feature's distribution is Gaussian[4]. Therefore, this section plans to nullify the following hypothesis:

3 It is noted that energy resembles a similar feature histogram.
4 Gaussian or normal distributions are usually represented by a mean and a standard deviation.

Case Processing Summary

	Cases					
	Valid		Missing		Total	
	N	Percent	N	Percent	N	Percent
Spectral Centroid	938480	100.0%	0	0.0%	938480	100.0%
Spectral Rolloff	938480	100.0%	0	0.0%	938480	100.0%
Spectral Energy	938480	100.0%	0	0.0%	938480	100.0%

Figure 6: Case processing summary before sample reduction.

Case Processing Summary

	Cases					
	Valid		Missing		Total	
	N	Percent	N	Percent	N	Percent
Spectral Centroid	5000	100.0%	0	0.0%	5000	100.0%
Spectral Rolloff	5000	100.0%	0	0.0%	5000	100.0%
Spectral Energy	5000	100.0%	0	0.0%	5000	100.0%

Figure 7: Case processing summary after sample reduction

Hypothesis 5.3. *Gaussian Frequency Distribution ($\mathcal{N}_0$)*
The spectral centroid, rolloff, and flux feature frequency distributions all follow normal distributions within a 5% significance level.

As a small visual aid, Figure 8 shows the frequency distribution of energy[5] binned to a 33-bin feature histogram. Superimposed on the feature histogram is a Gaussian distribution to represent energy's feature distribution. The Gaussian distribution is presented with only two constants: the mean, denoted μ, and standard deviation, denoted σ. Although it might seem suitable to represent energy's distribution by a Gaussian curve representation, we need to be sure that this is the best representation we can achieve. There exists several methods to test if a distribution is normal within a significance level, such methods include the Kolmogorov-Smirnov and Shapiro-Wilk test. The Kolmogorov-Smirnov test is a nonparametric test

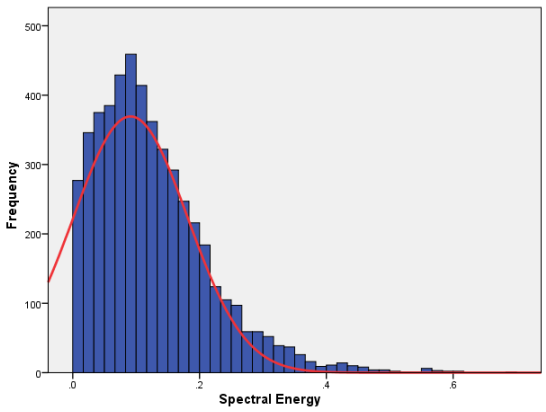


Figure 8: Energy of all GTZAN genres represented by a Gaussian distribution with $\mu = 0.091$ and the $\sigma = 0.09$.

5 The values of energy where extracted from GTZAN genres.

that is able to check if a feature's cumulative distribution function (cdf) is equal to the Gaussian cdf.

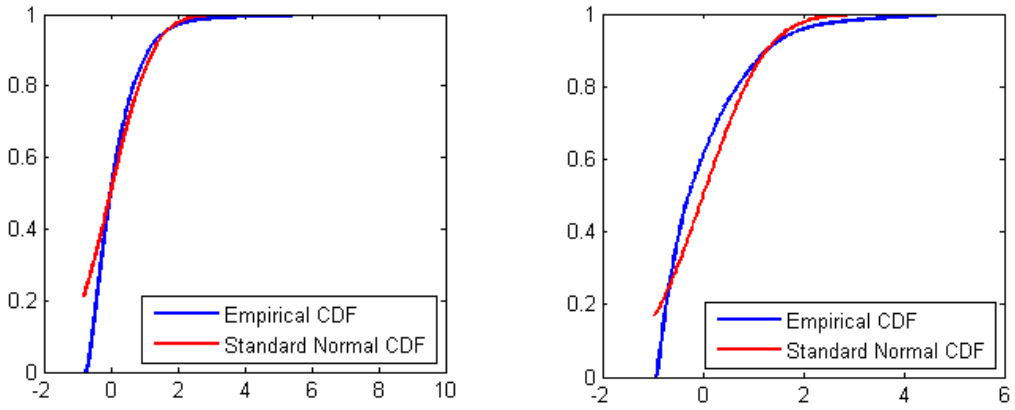
Figure 9 shows the results of using two tests of normality: Kolmogorov-Smirnov and Shapiro-Wilk. In the figure the Kolmogorov-Smirnov and Shapiro-Wilk rejects the significance (Sig.) for normality since all of the features have a significance less than 0.05 (5%).

	Kolmogorov-Smirnov[a]			Shapiro-Wilk		
	Statistic	df	Sig.	Statistic	df	Sig.
Spectral Centroid	.122	5000	.000	.834	5000	.000
Spectral Rolloff	.156	5000	.000	.830	5000	.000
Spectral Energy	.083	5000	.000	.912	5000	.000

a. Lilliefors Significance Correction

Figure 9: Reduced test for normality.

This is evident when considering the empirical cumulative distribution function (cdf) and the standard normal cdf in Figure 10 for spectral centroid and rolloff respectively[6]. In the figure the empirical cdf for both features do not precisely follow the standard normal cdf.



(a) ECDF for Centroid

(b) ECDF for Rolloff

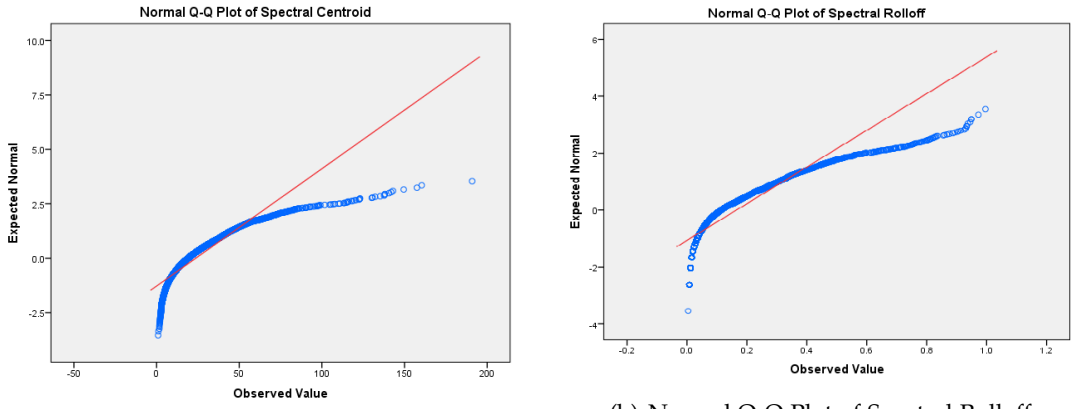
Figure 10: The plot shows the similarity between the empirical cdf of the centered and scaled feature vectors (centroid and rolloff) and the cdf of the standard normal distribution.

5.2.1 *Discussion and Conclusion*

We conclude by rejecting the null hypothesis 5.3 ($\mathcal{N}_0$), as the features: centroid, rolloff and energy frequency distributions do not follow normal distributions within a 5%

6 The ecdf and standard normal cdf for energy is similar.

significance level for both normality tests. It can be further shown that $\mathcal{N}_0$ is rejected by presenting the Q-Q plots for centroid and rolloff features[7]. Figure 11 shows the feature distribution with the plot symbol 'o'. Superimposed on the plot is a line joining the first and third quartiles of the graph, together these plots are referred to as Q-Q plots. These plots are constructed to graphically show if the feature distribution fits a normal distribution. If the sample is normal then the 'o' marked trend will follow the (red) line, this would indicate that the data is normally distributed. If the data is not Gaussian then the data will contain many curves and twists similar to those in Figure 11.



(a) Normal Q-Q Plot of Spectral Centroid

(b) Normal Q-Q Plot of Spectral Rolloff

Figure 11: Normal Q-Q plot for centroid and rolloff features.

5.3 OTHER FEATURE REPRESENTATIONS

In Section 5.2 we have shown that the features: spectral centroid, rolloff and energy do not follow normal distribution within a 5% significance level, therefore the mean and variance appear inadequate representations of these features and other techniques must be considered. In addition to the mean, the following feature representations will be explored by this dissertation for music genre classification:

Feature Histogram: The feature histogram arranges the feature's local window intensities into bin ranges. The content of each bin is counted and modelled by a frequency histogram. The histogram bin values are usually normalised and used for classification.

MFCC Aggregation: MFCC representation is a well-known feature representation which takes the first n MFC coefficients of the feature samples as it would a 16khz signal. If the feature contains more than one dimension, then each dimension is assessed independently and n coefficients will be produced per dimension. In this dissertation we took n = 4 as the number of MFC coefficients.

7 It is noted that the Q-Q plot for energy is similar.

See Fujinaga [1996]; McKay *et al.* [2005a] for some examples of how this method can be successfully used for genre classification.

Area Moments: Image moments is a central concept in computer vision and has its root in image processing. Fujinaga [1996] produced 10 such moments for image processing[8]. The ideas from Fujinaga [1996] were adapted for signal processing by McKay *et al.* [2005a] who created an algorithm that calculates seven moments using the original algorithm by Fujinaga [1996]. The representation in this research will employ the method used by McKay *et al.* [2005a].

After extracting the centroid, rolloff and energy features with two representations: one with the arithmetic mean and the other with the feature histogram, we achieved 37.3% for the feature histogram and 45.5% for the arithmetic mean and standard deviation using random forest classification with 10-fold cross validation. There are two main factors that should influence our choice of representation: a choice of dimensionality and classification precision. Although, in many cases, the 20-bin feature histogram will present better classification accuracy, one may argue that because of this dramatic increase in dimensionality, for this representation, the possibility of classification fits increase along with the algorithm's overall complexity. This could affect classification precision in the long run. Therefore, the best representation is one with minimal dimensionality increase that gives better classification precision. Figure 12 shows the effects of using the same features and classification algorithm with different feature representations. Figure 12(a) uses the 20-bin histogram, whereas Figure 12(b) uses the mean representation, were the row and column labels represent genre labels: G_1 = Blues, G_2 = Classical, G_3 = Country, G_4 = Disco, G_5 = Hiphop, G_6 = Jazz, G_7 = Metal, G_8 = Pop, G_9 = Raggae, G_{10} = Rock.

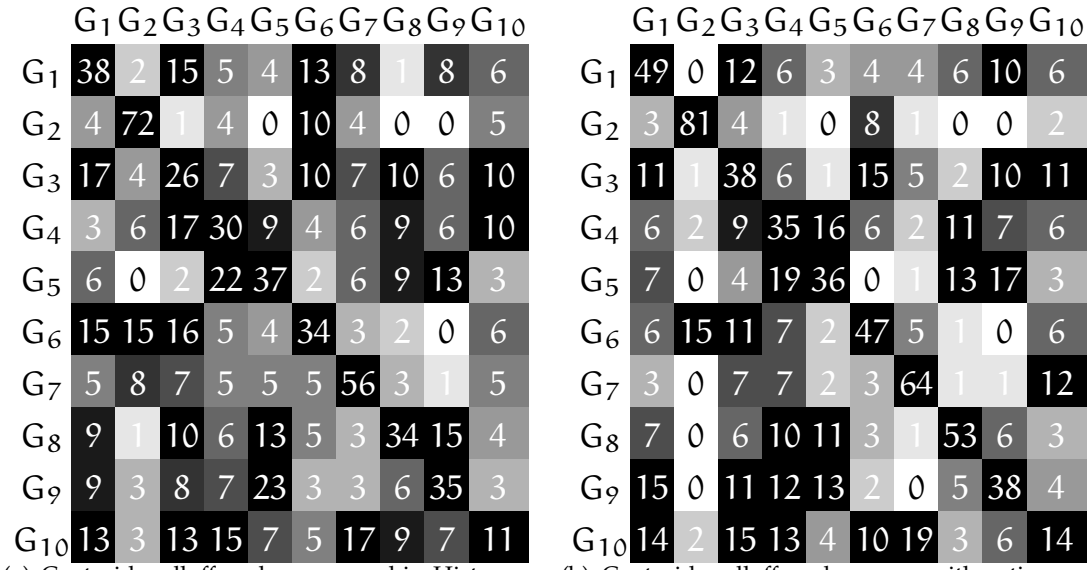
5.4 CONCLUSION AND DISCUSSION

Although the spectral centroid, rolloff and energy design matrices did not follow a Gaussian distribution within a 5% significance level, the mean representation still outperformed the 20-bin feature histogram in Figure 12. This result should not surprise us as rejecting null hypothesis $\mathcal{N}_0$ only tells us that the test failed to accept that the feature distributions followed normal distributions. Generally, this means that we can show that the feature distributions are not normally distributed, but we cannot show that they are.

Remark 5.4. Although, the rejection of $\mathcal{N}_0$ tells us that the feature distribution is not normally distributed, the tests cannot tell us if the distribution is skewed, fat-tailed, long tailed, heavy tailed, thin-tailed, e.t.c.

Furthermore, these normality tests rely heavily on the number of samples in the distribution to create these cdfs. If the sample is too small then the central limits

8 An image is treated as a 2-dimensional function $f(x, y) = z$, where x and y are indexes of the underlying matrix.



(a) Centroid, rolloff and energy 20-bin Histogram Classification

(b) Centroid, rolloff and energy arithmetic mean and standard deviation classification

Figure 12: Confusion matrices for 20-bin feature histogram and mean respectively.

theorem will cause an issue, whereas if there are too many samples in the distribution, unimportant digressions in the distribution will cause the normality test to fail. Therefore, for every feature described, multiple feature representations will be used to describe the feature distribution and based on this, along with dimensionality and classification precision, a feature representation will be selected.

6

MAGNITUDE BASED FEATURES

The magnitude spectrum, obtained from the fast Fourier transform of a signal, houses a family of spectral features for genre classification. Exploration of the magnitude spectrum has allowed us to identify signal change, noisiness, loudness and many other spectral features that describe aspects of discrete time signals for automatic music genre classification. Exploring peak-based features, from the local maxima of the frequency domain, creates opportunities to analyse the signal more thoroughly. In this chapter we explore the magnitude spectrum and present a compact design matrix to classify a benchmark dataset. Using only magnitude spectrum features we achieve 77.8% precision on 10 GTZAN genres using linear logistic regression models for automatic classification.

6.1 INTRODUCTION

The Fourier transform converts time series data to the frequency domain. The frequency domain is presented as a set of complex numbers, this can be very difficult visualise and so modern techniques represent this information in terms of two types of spectra: the *magnitude spectrum* and the *phase spectrum*. The magnitude spectrum is a collection of the magnitude of each complex number given by the frequency domain, equivalently the magnitude is the absolute value of the fast Fourier transform. The phase spectrum is made up of the angles (radians) of each complex component of the Fourier transform. In this chapter, we present a diverse set of descriptors obtained from the magnitude spectrum of a DTS to attempt to describe the indefinable nature of musical genre. Figure 13 shows an example of a magnitude spectrum obtained from "Arabesque No. 1" by Claude Debussy.

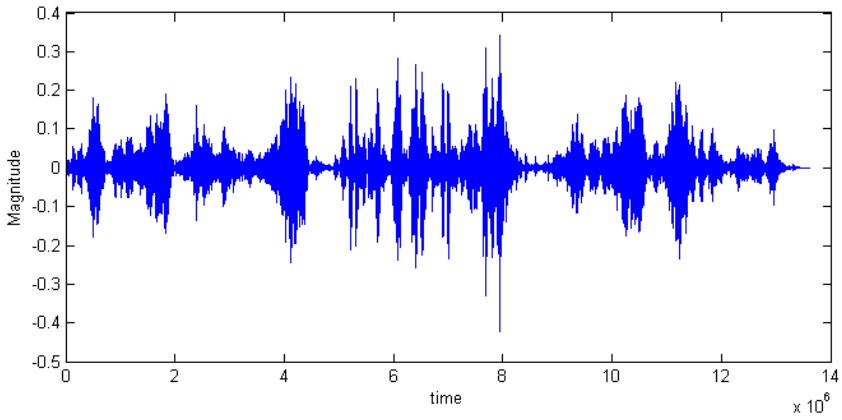


Figure 13: The magnitude spectrum of "Arabesque No. 1" by Claude Debussy.

In order to deduce the best representation of sound for a particular feature, we use the mean; MFCC representation; area methods; and a 20-bin feature histogram for

sound representation[1]. Thereafter, multiple classification techniques are performed to thoroughly assess the data. Six different classification algorithms are used in this analysis: K-Nearest Neighbours (KNN); Multilayer Perceptron (MP); Naïve Bayes (NB); Random Forests (RF); Linear Logistic Regression Models (LLRM); and Support Vector Machines (SVM). This chapter contains three main components: *The Magnitude Spectrum*, Section 6.2, provides a detailed feature analysis conducted using only features derived from the magnitude spectrum of a DTS; *The Power Cepstrum*, Section 6.3, extends the magnitude spectrum into the power ceptrum for feature descriptors for music genre classification; and finally in Section 6.4, a summary of these features are provided for music genre classification using magnitude-based *(including power spectrum)* features.

6.2 THE MAGNITUDE SPECTRUM

The are many different types of content-based features that can be derived from the magnitude spectrum. These content-based features tell us about spectral change (flux), noisiness (compactness) and many other aspects of music.

Remark 6.1. Although these descriptions might seem vague, signal flux and compactness are present in almost all types of music genres and must be considered for genre classification.

The definition for the magnitude spectrum is given by the following:

Definition 6.2. *Magnitude Spectrum*
Given a real and decaying (b>0) exponential signal:

$$x(t) = ae^{-bt}u(t), \tag{2}$$

the absolute value of the fast Fourier transform of the signal will yield the magnitude spectrum. Therefore, the magnitude spectrum can be computed as:

$$|\mathrm{FFT}(x(t))| = \mathrm{FFT}\left(\frac{a}{(b+jw)}\right) = \frac{|a|}{(b^2+w^2)^{\frac{1}{2}}}. \tag{3}$$

Now that we have a definition for the magnitude spectrum we can define a family of 13 content-based features for music genre classification: 9 from the magnitude spectrum and 4 from the power spectrum[2]. The 9 magnitude features include: *Spectral Slope*, Section 6.2.1, which describes whether a piece of music contains less energy at high frequencies; *Compactness*, Section 6.2.2, describes the noisiness of a signal; *Spectral Decrease*, Section 6.2.3, describes the degree to which there are more low frequency sounds to high frequency sounds in a DTS; *Loudness*, Section 6.2.4, describes total loudness in the bark band - bands produced by the bark scale which consist of the first 24 critical bands of human hearing - in terms of perceptual sharpness

1 These feature representations have been further expanded in Chapter 5.
2 The power spectrum is the absolute square of the magnitude spectrum.

(Section 6.2.4.1) and spread (Section 6.2.4.2); *Onset Detection*, Section 6.2.5, which describes how a piece of music starts by describing the rise in magnitude from zero; *Octave Band Signal Intensity*, Section 6.2.6, using a triangular octave filter bank; *Peak-based features*, Section 6.2.7, a family of features based upon the peaks of the DTS; *Spectral Flux*, Section 6.2.8, measures the rate of change of the magnitude spectrum; and *Spectral Variability*, Section 6.2.9, which measures how closely or spread-out the signal is clustered. McKay and Fujinaga [2006] explained the significant link between musicology and psychology, he maintains that this link can be exploited for genre classification. Therefore, this section is presented in a way to explain the musicological and psychological aspects that are involved in each suggested feature. Some of the underlying mathematical concepts are given in the Appendix A section of this dissertation.

6.2.1 *Spectral Slope*

The *spectral slope* for a continuous natural signal has been understood for many years [Fry 1979]. The spectral slope can be observed when natural audio signals tend to have less energy at high frequencies. Peeters [2004] provides a way to quantify this by applying a linear regression to the *magnitude spectrum* of the signal, which produces a single number indicating the slope of the line-of-best-fit through the spectral data. Spectral slope is just one feature that uses energy distribution over the frequency of the DTS, other features that use this property include spectral rolloff and centroid [Peeters 2004].

Using spectral slope represented by the mean we achieve accuracies of 24.4% using naïve Bayes; 24.8% using support vector machines; 24% using the multilayer perceptron; 24.4% using linear logistic regression models; 18.3% using k-nearest neighbours; and 19.3% using random forests with 10-fold cross validation on GTZAN genres. Although spectral slope does not seem to distinguish genres well, this is because similar spectral slope is sometimes shared between genres. However, we can expect music that rely heavily on high pitched melodies to have distinguishing spectral slope. For this reason, disco and classical music will often have dissimilar spectral slope compared to other genres.

6.2.1.1 *Strongest Frequency*

The fast Fourier transform provides an excellent representation of sound and so acknowledging simple aspects of this transform can be very fruitful.

Example 6.3. For example, taking the strongest frequency of the FFT, in Hz, can also be a good feature to measure if a particular genre is bounded by a maximum frequency strength.

Table 5 shows the classification scores of the strongest frequency using three representations with multiple classifiers. Using the strongest frequency, represented by MFCCs, we achieve accuracies of 23.6% using naïve Bayes; 21.9% using support vector machines; 23.4% using the multilayer perceptron; 24.5% using linear logistic regression models; 20% using k-nearest neighbours; and 23.8% using random forests

Representation	KNN	MP	NB	RF	LLRM	SVM
Mean	23.40%	24.90%	18.60%	25.20%	23.00%	14.30%
MFCC	20.00%	23.40%	23.60%	23.80%	24.50%	21.90%
3-bin FH	18.80%	18.60%	16.70%	16.90%	18.20%	15.80%
5-bin FH	20.90%	22.10%	21.00%	22.10%	21.00%	18.40%
10-bin FH	21.60%	24.30%	22.00%	22.30%	24.70%	21.00%
20-bin FH	22.70%	25.40%	23.60%	24.30%	24.90%	22.50%
30-bin FH	23.30%	24.60%	21.90%	25.60%	24.30%	22.30%

Table 5: Classification scores for different feature representations for FFT maximum using a variety of classification techniques.

with 10-fold cross validation on GTZAN genres. While the feature histogram representation provides slightly better classification, increasing the dimensionality and thus computation, by using a 30-bin feature histogram just for a 0.6% increase in classification overall, does not warrant the improved performance.

As shown in Figure 14, the strongest frequency cannot independently provide a good feature for genre classification as music genre constitute many aspects. However some sub-genres, such as 'Furniture music' that falls part of classical music, maintains very weak frequencies, whereas other sub-genres such as 'Baroque' (also early classical) maintain very strong frequencies. Therefore, although the GTZAN dataset does not provide an extensive flavour of musical genre, we still should expect enormous real databases to contain this diversity of genre and so these detailed features should also be considered.

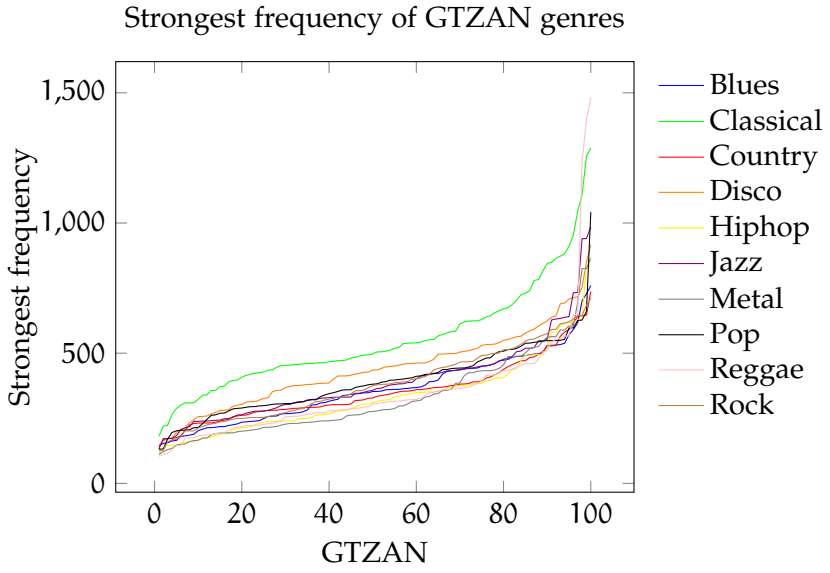


Figure 14: Strongest frequency feature values for 10 GTZAN genres using the mean representation.

6.2.2 *Compactness*

Compactness is a measure of the noisiness of a signal [McKay *et al.* 2005a] and is calculated by comparing the value of a magnitude spectrum bin with its surrounding values. In many genres (e.g. metal) a random and persistent disturbance that obscures the clarity of sound is desired, which this feature will detect.

Representation	KNN	MP	NB	RF	LLRM	SVM
Mean	33.50%	39.30%	38.30%	35.60%	40.70%	24.70%
MFCC	26.30%	32.50%	32.10%	28.70%	32.90%	33.60%
20-bin FH	20.50%	21.30%	19.40%	20.60%	21.50%	17.30%
Area Moments	25.5%	27.10%	27.40%	27.70%	26.10%	12.20%

Table 6: Classification scores for different feature representations for compactness using a variety of classification techniques.

Figure 15 shows the compactness feature values of 10 GTZAN genres with one hundred 30 sec excerpts for each genre. The figure shows the range of values for compactness for each genre represented by the mean. Although, in some areas in the figure, these genres overlap indicating that the excerpts correspond to the same feature value which could cause possible misclassification of genre, using this figure we can expect a classical piece of music to have an average compactness between 1.700-1.900. Since most of the other genres are out of this moving range we should expect the correct classification of classical music to be fairly high.

The compactness of a signal is represented using several sound representation techniques outlined in Chapter 5: using the MFCC representation we achieve accuracies of 32.9% correctly classified genres; using the mean representation we achieved 40.7%; using a 20-bin feature histogram we achieve 21.5%; and finally using area methods of moments we achieve 26.1% accuracy on 10 GTZAN genres using linear logistic regression models with 10-fold cross validation. These statistics suggest that compactness using the mean representation is a worthy classification feature for classifying 10 genres, doing better than a $\frac{1}{10} = 10\%$ chance, and overall produces very good classification scores for blues, classical, jazz and hiphop genres.

6.2.3 *Spectral Decrease*

Spectral decrease can be used to distinguish instruments used in music signals. For this reason, spectral decrease can be used to identify instruments which are unusually used in a particular genre.

Example 6.4. For example, classical music will have more high frequency sounds (use of violins), contrastingly, contemporary music will have many low frequency sounds (use of a double bass guitars and kick drums rather than an acoustic guitar).

Peeters [2003] used spectral decrease to correctly classify large musical instrument databases. The definition for spectral decrease is given as follows:

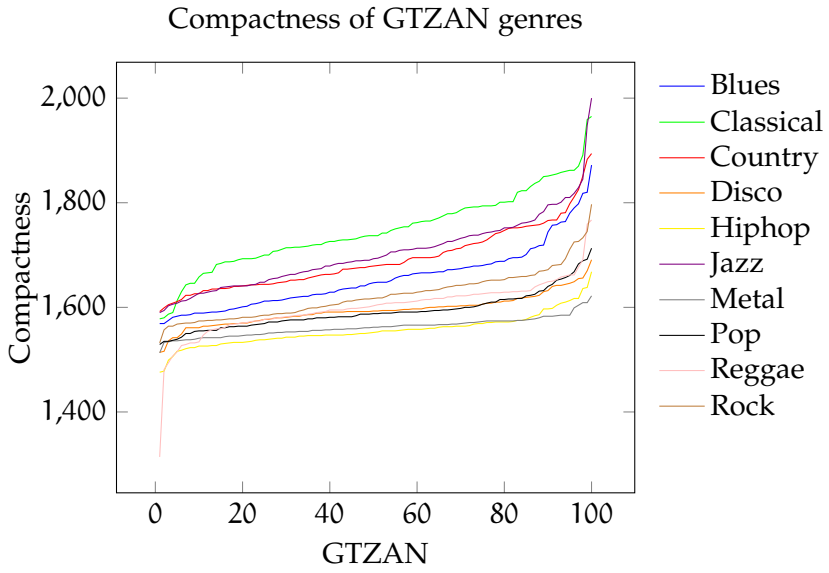


Figure 15: Compactness feature values for 10 GTZAN genres using the mean representation.

Definition 6.5. *Spectral Decrease*
Spectral decrease is a feature which measures the degree to which there are more low frequency sounds than high frequency sounds. This feature is a ratio that will be large if there are more low frequency sounds than high frequency sounds.

Using spectral decrease, represented by the mean, we achieve accuracies of 20.8% using naïve Bayes; 17.3% using support vector machines; 20.9% using the multi-layer perceptron; 19.8% using linear logistic regression models; 19.5% using k-nearest neighbours; and 20.1% using random forests with 10-fold cross validation on GTZAN genres. These are not particularly high classifications since most genres share similar instrumentation but differ in harmony and scale use. Therefore, other features also need to explore these aspects as well.

6.2.4 *Loudness*

Specific loudness is the loudness associated with each bark band [Peeters 2004; Moore *et al.* 1997; Wold *et al.* 1996], and is denoted by $N'(z)$, where z is the z^{th} frequency in the bark band. Total loudness has been used for multi-speaker speech activity detection [Pfau *et al.* 2001], automatic speech recognition [Zwicker *et al.* 1979; Reichl and Ruske 1995 2011], instrument recognition [Essid *et al.* 2006] and music genre classification [Benetos and Kotropoulos 2008]. Before the definition for total loudness can be given, the definition for specific loudness must first be defined upon local intervals on the audio signal. Rodet and Tisserand [2001] presented an approximation of specific loudness for an audio signal by expressing it in its relative scale in terms of energy:

$$N'(z) = E(z)^{0.02}. \tag{4}$$

Zwicker and Fastl [1990] extended this definition for total loudness which is the total sum of specific loudness:

$$N = \sum_{z=1}^{\text{ALL BANDS}} N'(z), \tag{5}$$

where $N'(z)$ is the specific loudness for the z^{th} band. Further work by Moore *et al.* [1997] gives a more precise definition for total and specific loudness. The loudness feature can be further extended to include *perceptual sharpness* and *perceptual spread*.

6.2.4.1 *Perceptual Sharpness*

According to Zwicker [1977], the sharpness of an audio signal is perceptually equivalent to the spectral centroid but computed using the specific loudness of the bark bands. Perceptual sharpness is used for video compression [Yang *et al.* 2006], blur detection [Narvekar and Karam 2009] and image resolution enhancement [Liu 1999]. Using Equation 6.14 and Peeters [2004] the perceptual sharpness is given as:

$$A = 0.11 \cdot \frac{\sum_{z=1}^{allbands} z \cdot g(z) \cdot N'(z)}{N}, \tag{6}$$

where z is the index of the band and $g(z)$ is a function defined by:

$$g(z) := \begin{cases} 1 \text{ if } z < 15 \\ 0.066 \cdot \exp(0.171z) \text{ if } z \geqslant 15. \end{cases} . \tag{7}$$

6.2.4.2 *Perceptual Spread*

The perceptual spread calculates the distance from the largest specific loudness value to the total loudness. Therefore, an equation for perceptual spread is given as:

$$ET = (\frac{N - max_z N'(z)}{N})^2. \tag{8}$$

Perceptual spread has been used for watermarking multimedia [Cox *et al.* 1997], image classification [Ahumada Jr 1996], and image watermarking [Kankanhalli and Ramakrishnan 1998].

Using *total loudness*, *perceptual sharpness*, and *perceptual spread*, represented by the mean, we achieve accuracies of 45.7% using naïve Bayes; 26.1% using support vector machines; 54.8% using the multilayer perceptron; 56.6% using linear logistic regression models; 50.3% using k-nearest neighbours; and 52% using random forests with 10-fold cross validation on GTZAN genres, making this a worthy feature for genre classification.

6.2.5 *Onset Detection*

Duxbury *et al.* [2003] provided a method to compute onset detection using complex domain spectral flux. Onset detection describes information about the initial magnitude of a piece of music. This feature describes the rise in magnitude from zero to some initial value. Using *complex domain onset detection*, represented by the mean, we achieve accuracies of 25.8% using naïve Bayes; 21.5% using support vector machines; 26.3% using the multilayer perceptron; 25.5% using linear logistic regression models; 22% using k-nearest neighbours; and 22.9% using random forests with 10-fold cross validation on GTZAN genres. Onset detection classifies pop, classical, and jazz well.

6.2.6 *Octave Band Signal Intensity*

Essid [2005], provides a way to compute octave band signal intensity using a trigular octave filter bank. We can also compute the log of OBSI ratio between consecutive octaves. Using *octave band signal intensity* and *octave band signal intensity ratio*, represented by the mean, we achieve accuracies of 47.9% using naïve Bayes; 55.7% using support vector machines; 49.1% using the multilayer perceptron; 48.8% using linear logistic regression models; 52.6% using k-nearest neighbours; and 52.4% using random forests with 10-fold cross validation on GTZAN genres. OBSI and OBSIR are very good classification features across most genre types.

6.2.7 *Peak Detection*

Studying the peaks of a signal allows us to account for various principal features that are contained within a signal. For example, peak-based features such as crest factor, peak flux, centroid and smoothness can help us describe the quality of AC waveform power and detecting vibration. Music recordings have very widely fluctuating peak-based features, therefore exploiting these characteristics can be fruitful [Helen and Virtanen 2005]. The peak detection algorithm by McKay *et al.* [2005b] will be used for extracting peak-based features. McKay *et al.* [2005b] calculated peaks by detecting local maximums in the frequency bins, these maxima are calculated within a threshold where the largest maxima within this threshold is considered. These global peaks per threshold[3] are considered without any information about its bin location. Treating this set of peak values together as a 16khz signal we can attempt to represent this by centroid, flux and smoothness.

6.2.7.1 *Peak Centroid*

Peak centroid is calculated from the peak set extracted from an audio signal. These values were constructed following the implementation by Peeters *et al.* [2000].

3 In our experiments we took a peak threshold of 10.

Representation	KNN	MP	NB	RF	LLRM	SVM
Mean	29.10%	33.60%	28.60%	30.60%	32.50%	35.00%
MFCC	25.30%	30.60%	28.40%	29.50%	28.30%	32.30%
20-bin FH	25.90%	28.00%	26.40%	31.50%	29.70%	23.60%
Area Moments	21.00%	23.10%	22.40%	20.10%	20.20%	14.90%

Table 7: Classification scores for different feature representations for peak centroid using a variety of classification techniques.

Using *peak centroid*, represented by the mean, we achieve accuracies of 28.6% using naïve Bayes; 35% using support vector machines; 33.6% using the multilayer perceptron; 32.5% using linear logistic regression models; 29.1% using k-nearest neighbours; and 30.6% using random forests with 10-fold cross validation on GTZAN genres. Figure 16 shows the peak centroid feature values of 10 GTZAN genres. The figure shows the range of values for peak centroid for each genre represented by the mean. Using this figure we can expect a metal piece of music to have an average value between 2.5-4.5 for peak centroid. Since most of the other genres are out of this moving range we should expect the correct classification of metal music to be fairly high.

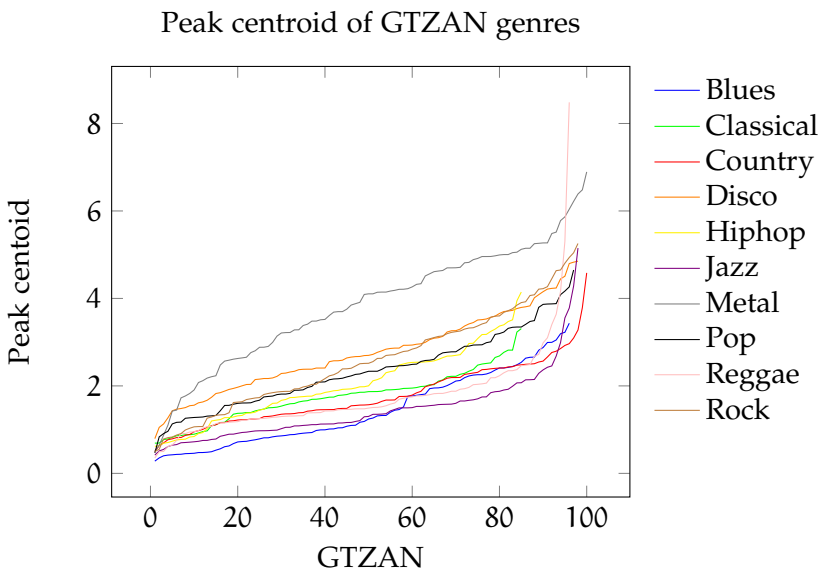


Figure 16: Peak centroid feature values for 10 GTZAN genres using the mean representation.

6.2.7.2 *Peak Flux*

Similarly, peak flux is also calculated from a peak value set. The extraction methodology followed that of McKay *et al.* [2005b]; Peeters *et al.* [2000], where the correlation between adjacent peaks are considered. Using *peak flux*, represented by a 20-bin feature histogram, we achieve accuracies of 14.7% using naïve Bayes; 15% using support vector machines; 20.1% using the multilayer perceptron; 18.7% using linear logistic regression models; 17.2% using k-nearest neighbours; and 20.5% using random forests with 10-fold cross validation on GTZAN genres.

Representation	KNN	MP	NB	RF	LLRM	SVM
Mean	11.30%	18.00%	14.60%	14.40%	17.60%	14.10%
MFCC	13.30%	17.50%	13.10%	12.30%	17.00%	17.40%
20-bin FH	17.20%	20.10%	14.70%	20.50%	18.70%	15.00%
Area Moments	14.40%	17.60%	14.30%	14.20%	15.40%	17.00%

Table 8: Classification scores for different feature representations for peak flux using a variety of classification techniques.

6.2.7.3 *Spectral Crest Factor*

A musical discrete-time signal[4] has a widely varying crest factor. The crest factor is a feature of a DTS which displays the ratio of peak points to the RMS, and is measured in decibels (dB). The spectral crest factor shows the magnitude of peaks in the spectrum of the DTS. A spectral crest factor of 1 tells us that the spectrum of the DTS has no peaks, contrastingly, a much larger spectral crest factor will describe many peaks. Common values of a spectral crest factor for an audio mix[5] are around 4-8dB, and 8-10dB for an unprocessed recording. Jang *et al.* [2008] used spectral crest factor per band to classify music genre.

Definition 6.6. *Spectral Crest Factor*
The spectral crest factor is measured by taking the peak amplitude and dividing it by the RMS of the DTS.

$$\mathrm{CrestFactor} = \frac{\max(x)}{x_{\mathrm{RMS}}}. \tag{9}$$

Using *spectral crest factor*, represented by the mean, we achieve accuracies of 41.1% using naïve Bayes; 47.8% using support vector machines; 46.1% using the multi-layer perceptron; 49.5% using linear logistic regression models; 44.4% using k-nearest neighbours; and 45% using random forests with 10-fold cross validation on GTZAN genres. Crest factor is a very powerful feature for genre classification, however, this feature is represented by 19 components, and therefore 19 dimensions. On the other hand, crest factor is particularly useful to correctly identify classical, pop and metal genres.

Representation	KNN	MP	NB	RF	LLRM	SVM
Mean	27.50%	32.90%	28.00%	29.20%	31.00%	25.60%
MFCC	23.30%	30.60%	28.90%	28.30%	28.80%	29.80%
20-bin FH	25.20%	27.40%	25.50%	27.10%	27.80%	22.10%
Area Moments	21.50%	20.80%	19.00%	21.10%	18.20%	12.20%

Table 9: Classification scores for different feature representations for peak smoothness using a variety of classification techniques.

4 A discrete-time signal obtained from composed music.
5 Multiple recorded sounds are combined into one or more channels.

6.2.7.4 *Peak Smoothness*

Finally, peak smoothness is calculated from the peak value set by evaluating the log of a peak subtracted from the log of the surrounding peaks [McKay *et al.* 2005b]. Smoothing out peak values allows us to describe inconsistent or unexpected rises in amplitude that could occur in classical and metal genre. Using *peak smoothness*, represented by the mean, we achieve accuracies of 28% using naïve Bayes; 25.6% using support vector machines; 32.9% using the multilayer perceptron; 31% using linear logistic regression models; 27.5% using k-nearest neighbours; and 29.2% using random forests with 10-fold cross validation on GTZAN genres.

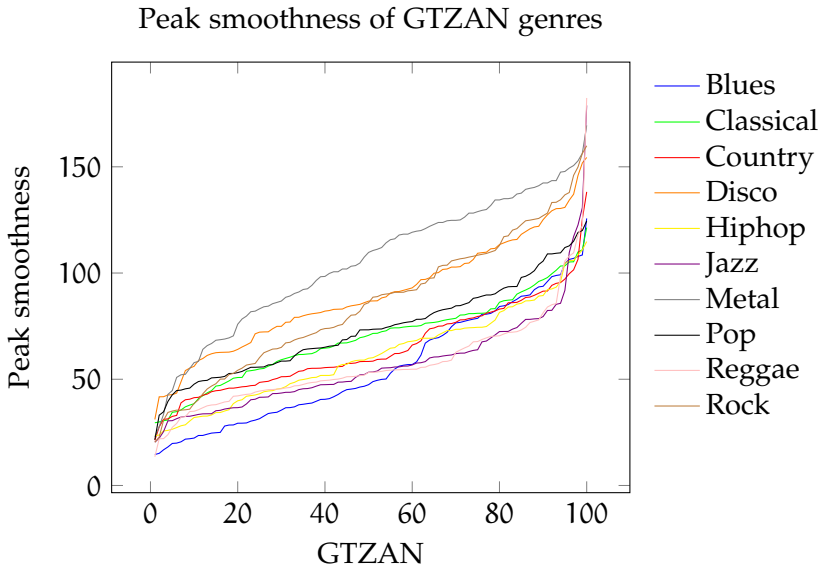


Figure 17: Peak smoothness feature values for 10 GTZAN genres using the mean representation.

6.2.8 *Spectral Flux*

Spectral flux is a content-based feature that measures the rate of change of the magnitude spectrum for the DTS. This is achieved by comparing every frame of the magnitude spectrum with its previous frame [Giannoulis *et al.* 2012]. The spectral flux is used for onset detection [Dixon 2006] and audio classification [Tzanetakis and Cook 2002; Lu *et al.* 2002].

Definition 6.7. *Spectral Flux*
The spectral flux is computed by taking the normalised product between two successive normalised amplitude spectra: $a(t-1)$ and $a(t)$ [Peeters 2004]. Then the spectral flux is given as:

$$S_{flux} = 1 - \frac{\sum_k a(t-1,k).a(t,k)}{\sqrt{\sum_k a(t-1,k)^2}\sqrt{\sum_k a(t,k)^2}}. \tag{10}$$

It can be shown that

$$0 \leqslant S_{flux} \leqslant 1. \tag{11}$$

The interpretation of the spectral flux is given as:

$$\begin{cases} \text{IF } S_{flux} \to 0 \text{ THEN THE SUCCESSIVE SPECTRA ARE SIMILAR.} \\ \text{IF } S_{flux} \to 1 \text{ THEN THE SUCCESSIVE SPECTRA ARE DISSIMILAR.} \end{cases} \tag{12}$$

Representation	KNN	MP	NB	RF	LLRM	SVM
Mean	26.00%	32.90%	24.80%	27.20%	29.50%	18.20%
MFCC	30.20%	34.80%	35.90%	30.80%	36.40%	36.00%
20-bin FH	20.10%	21.10%	19.40%	22.70%	21.70%	17.20%
Area Moments	28.30%	25.00%	23.90%	30.40%	26.30%	25.60%

Table 10: Classification scores for different feature representations for spectral flux using a variety of classification techniques.

Table 10 shows spectral flux represented using several sound representation techniques: using the MFCC representation we achieve accuracies of 36.4% correctly classified genres, using the mean representation we achieved 32.9%, using a 20-bin feature histogram we achieve 21.7%, and finally using area methods of moments we achieve 30.4% accuracy on GTZAN genres using 10-fold cross validation and multiple classification techniques. It would seem that the MFCC representation is preferred by spectral flux and performs the best when classified with linear logistic regression models.

6.2.9 *Spectral Variability*

Statistical variability measures dispersion in data, i.e. how closely or spread-out the signal is clustered. We can achieve this my measuring the standard deviation of the magnitude spectrum of the signal. Figure 18 shows the classification scores of 10 GTZAN genres with one hundred 30 sec excerpts for each genre. The figure shows the range of values for variability for each genre represented by the mean. Although in some areas in the figure these genres overlap indicating a possible misclassification of genres, using this figure we can expect a jazz piece of music to have an average variability between 0.1-0.3 ($.10^{-2}$), and therefore, we should expect the correct classification of jazz music to be fairly high. The same deduction can be expended for pop, classical, and hiphop, whereas other genres mostly stay in the same ranges which could cause misclassification.

Table 11 presents the classification scores for different feature representations for variability using a verity of classification algorithms. We can see that variability is best represented by the MFCC feature representation (from the available representations) and achieves 38.40% classification precision using Naïve Bayes.

6.3 THE POWER CEPSTRUM

The power cepstrum is defined as the rate of change in different spectrum bands. The tonal formants and pitch are additive in the log of the power spectrum, this makes

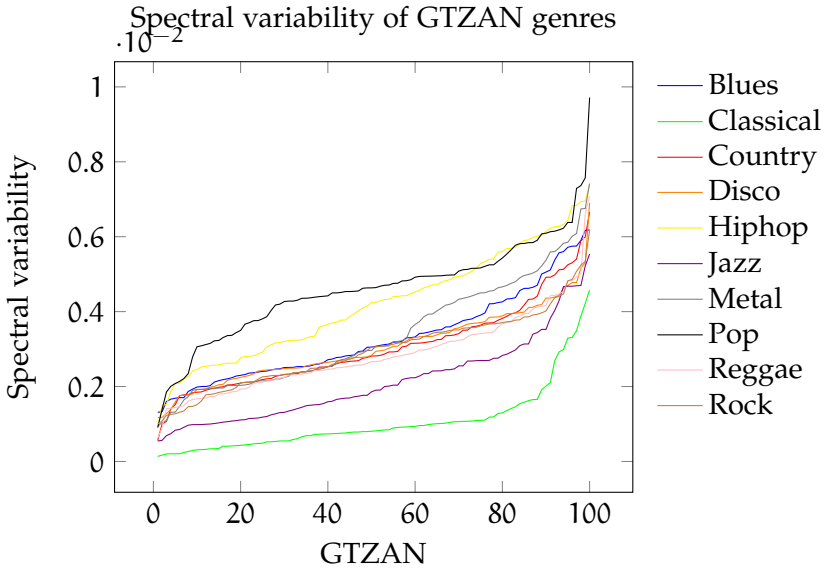


Figure 18: Spectral variability feature values for 10 GTZAN genres using the mean representation.

Representation	KNN	MP	NB	RF	LLRM	SVM
Mean	28.70%	32.30%	26.80%	31.00%	30.20%	21.40%
MFCC	31.50%	37.00%	38.40%	36.90%	36.40%	38.20%
20-bin FH	23.80%	23.80%	21.60%	23.90%	22.30%	17.60%
Area Moments	25.90%	26.70%	25.80%	28.30%	23.90%	24.00%

Table 11: Classification scores for different feature representations for spectral variability using a variety of classification techniques.

them clearly separate [Noll 2005a]. Therefore, this is an ideal tool for determining the frequency of human speech. Before further defining the power cepstrum, the definition of the Fourier transform for discrete signals must be reviewed.

Definition 6.8. *Discrete Fourier Transform*
The discrete-time Fourier transform, notated as $\mathcal{DFT}$, of a discrete-time real set, x(n) $\forall n \in \mathbb{Z}^+$, is defined by:

$$X_{2\pi}(\omega) = \sum_{n=-\infty}^{\infty} x(n)e^{-i\omega n}, \tag{13}$$

where ω (frequency variable) has normalised sample units and X is a periodic function with periodicity 2π.

Definition 6.9. *Power Cepstrum*
The power cepstrum of a signal is defined as the squared magnitude of the inverse $\mathcal{DFT}$ of the logarithm of the squared magnitude of the $\mathcal{DFT}$ of a DTS:

$$\text{POWER CEPSTRUM OF SIGNAL} = |\mathcal{DFT}^{-1}\{\log(|\mathcal{DFT}\{f(t)\}|^2)\}|^2. \tag{14}$$

A short-time cepstrum analysis was proposed by Noll and Schroeder [2005]; Noll [2005ba] for pitch determination of human speech. The power cepstrum is a commonly used feature for describing a musical DTS. This is done by converting the DTS, using the Mel (melody) scale to Mel-frequency cepstrum [6] (MFC). The MFC is used as a design matrix for identifying the human voice; musical signals; and content-based audio classification [Li 2000; Guo and Li 2003]. In the next two definitions, the discrete cosine transform II is used to further define the Mel-frequency cepstral coefficients (MFCCs).

Definition 6.10. *Discrete Cosine Transform II*[7]
The discrete cosine transform (denoted DCT), is defined as a linear, invertible function $f : \mathbb{R}^N \to \mathbb{R}^N$, or equivalently an invertible $N \times N$ square matrix. Let $N = \{x_0, ..., x_{N-1}\}$ be a set where $x_i \in \mathbb{R}\ \forall i \in \{0, ..., N-1\}$, then one can define the DCT of set N as:

$$X_k = \sum_{n=0}^{N-1} x_n \cos[\frac{\pi}{N}(n + \frac{1}{2})k]\ k = 0, ..., N-1. \tag{15}$$

The output set is $\bar{N}$ real numbers $\{X_0, ..., X_{N-1}\}$ which is the transformed set N.

Now that we have a definition for the power cepstrum we can extend the 9 features defined in the previous section with the 4 features from the power spectrum. The 4 power cepstrum features include: *Mel-Frequency Cepstral Coefficients*, Section 6.3.1, a powerful feature for genre detection and representation for musical signal features; *Spectral Flatness*, Section 6.3.2, which is used to measure how pure tonal sounds are

6 MFC is a representation of the short-term power spectrum of a sound, based on a linear cosine transform of a log power spectrum on a nonlinear Mel-scale of frequency
7 The second of four conventional definitions of DCT.

compared to noisy ones; *Spectral Shape Statistics*, Section 6.3.3, global descriptors of a DTS; *Spectral Rolloff*, Section 6.3.4, which measures the amount of right-skewedness of the power spectrum.

6.3.1 *Mel-Frequency Cepstral Coefficients*

Mel-frequency cepstral coefficients (MFCCs) are the coefficients that together make up an Mel-frequency cepstrum. The components of MFCC are those from the cepstral representation of the audio signal. In the Mel-frequency cepstrum the frequency bands are equally spaced which favours the human auditory system more than using the cepstrum feature alone, which uses linearly-spaced frequency bands. There are many more uses for MFC including signal compression.

Definition 6.11. *Mel-Frequency Cepstral Coefficients*
Xu *et al.* [2005b]; Sahidullah and Saha [2012] have presented five steps to acquire MFCCs from a DTS:

1. Take the discrete-time Fourier transform of the signal, using local windows.
2. Map the powers of the spectrum obtained above onto the Mel-scale, using triangular overlapping windows.
3. Take the logs of the powers at each of the Mel-frequencies.
4. Take the discrete cosine transform of the list of Mel log powers, as if it was a signal.
5. The MFCCs are the amplitudes of the resulting spectrum.

The shape or spacing of the local windows can be adjusted [Zheng *et al.* 2001] as well as cepstral/spectral dynamics features [Furui 1986]. The Mel-filter bank is built as 40 log-spaced filters according to the following Mel-scale conversion. The frequency, f (Hertz), to Mel, m, is given by the following conversion formula:

$$m = 1127 * \log(1 + \frac{f}{700}), \tag{16}$$

each filter is a triangular filter with hight $\frac{2}{(f_{max} - f_{min})}$. Then MFCCs are computed as follows, using the discrete cosines transform by Equation 14:

$$mfcc = dct(\log(abs(\textsc{fft}(hanning(N).x)).\textsc{MelFilterBank})), \tag{17}$$

where the hanning window is given in Appendix A. MFCCs have been used in music modelling [Logan 2000]; early classification of bearing faults [Nelwamondo *et al.* 2006]; comparisons of parametric representations for spoken sentences [Davis and Mermelstein 1980]; and finally in music genre classification [Cast *et al.* 2014].

The MFCCs can be represented in many different ways and is in itself also considered a powerful representation. In this study we considered 3 representations for MFCCs. Using the MFCC representation we achieve accuracies of 66.6%; using the mean representation we achieved 58.30%; and finally using area methods of moments

we achieve 37.20% accuracy on GTZAN genres using 10 fold cross validation. Table 12 outlines all precisions obtained using multiple classification algorithms. The best representation according to Table 12 is the MFCC representation obtaining 66.60% accuracy using support vector machines.

Representation	KNN	MP	NB	RF	LLRM	SVM
Mean	53.20%	55.90%	50.10%	49.40%	58.30%	52.90%
MFCC	50.30%	59.40%	53.90%	55.20%	61.70%	66.60%
Area Moments	32.20%	34.80%	29.00%	33.20%	37.20%	12.60%

Table 12: Classification scores for different feature representations for MFCCs using a variety of classification techniques.

6.3.2 *Flatness*

Spectral flatness is a feature used to calibrate how pure tonal sounds are in comparison to noisy ones [Dubnov 2004]. Pure tonal sound refers to resonant structure in a power spectrum[8], compared to other parts containing white noise. A high spectral flatness (approaching 1.0 for white noise) indicates that the spectrum has a similar amount of power in all spectral bands - this would sound similar to white noise, and the graph of the spectrum would appear relatively flat and smooth. A low spectral flatness (approaching 0.0 for a pure tone) indicates that the spectral power is concentrated in a relatively small number of bands - this would typically sound like a mixture of sine waves, and the spectrum would appear "spiky" [Peeters 2004]. Herre *et al.* [2001] and Jang *et al.* [2008] used flatness to match audio signals and classify genre respectively. Theorem 6.13 is presented below using two means described in Appendix A to define spectral flatness.

Definition 6.12. *Wiener Entropy or Spectral Flatness*
Spectral flatness is defined as the ratio of the geometric mean (Section A.3) to the arithmetic mean (Section A.2) of the DTS:

$$\text{SpecFlatness} = \frac{\text{GeoMean}(x)}{\text{Arith}(x)} = \frac{\sqrt[N]{\prod_{i=0}^{N-1} x(i)}}{\frac{\sum_{i=0}^{N-1} x(i)}{N}} = \frac{e^{(\frac{1}{N}\sum_{i=1}^{N-1} \ln x(i))}}{\frac{1}{N}\sum_{i=0}^{N-1} x(i)}, \tag{18}$$

where $x(n)$ represents the DTS. The spectral flatness feature is also useful as a local feature rather than a global one. By Theorem 6.13 the spectral flatness will always be a positive real number between 0 and 1 inclusive.

Theorem 6.13. *Let* $S = \{x_1, x_2, x_3, x_4, ..., x_n\}$. *Furthermore, let* $\bar{x}_A$ *and* $\bar{x}_G$ *be the arithmetic (Section A.2) and geometric mean (Section A.3) of* S *respectively. If* S *contains no pair of elements such that* $x_i = x_j \ \forall i, j \in \{1, ..., n\}$ *where* $i \neq j$, *then*

$$\bar{x}_G < \bar{x}_A. \tag{19}$$

8 The power spectrum of a DTS is the power of that DTS at each frequency that it contains

Using spectral flatness represented by the mean we achieve accuracies of 41.8% using naïve Bayes; 39.8% using support vector machines; 50.5% using the multilayer perceptron; 51.6% using linear logistic regression models; 49.5% using k-nearest neighbours; and 44.2% using random forests with 10-fold cross validation. These classification scores make spectral flatness a noteworthy descriptor of genre.

6.3.3 *Spectral Shape Statistics*

Spectral statistics are used as global descriptors of the DTS. There are four types of spectral shape statistical features: centroid, spread, kurtosis and skewness. The following definitions extends these statistical feature concepts in terms of the spectral centroid, however first the spectral centroid must be defined.

6.3.3.1 *Spectral Centroid*

The spectral centroid has been used to predict spectral "brightness" of a DTS [Grey and Gordon 1978] and is used widely in digital audio processing as a tool to measure musical timbre [Schubert *et al.* 2004]. The spectral centroid is commonly used to categorise audio signals [Tzanetakis and Cook 2002] and audio data [Li *et al.* 2001]. The brightness can be defined as the precise place where the "centre of spectral mass" exists on the DTS. According to Grey and Gordon [1978], this property has a robust connection with the impression of "brightness" of a sound.

Definition 6.14. *Spectral Centroid*
The spectral centroid is defined as the weighted mean Section A.5 of the frequencies present in the DTS, determined using the discrete-time Fourier transform, with their magnitudes as the weights [Peeters 2004]:

$$\text{Spectral Centroid} = [\mathbb{SC}] = \frac{\sum_{n=0}^{N-1} f(n)x(n)}{\sum_{n=0}^{N-1} x(n)}, \tag{20}$$

where $x(n)$ represents the weight (magnitude at that frequency) of bin number n, and $f(n)$ represents the center frequency of that bin.

Recall that spectral centroid refers to the centre of mass in the power spectrum and is often used to calculate the brightness of a music signal. This feature has been widely used in signal processing as a true timbre descriptor. The results in Figure 19 was calculated with a sample rate of 16khz over a 512 sample window size. Using the mean representation for spectral centroid we explored the sensitivity to change of the dependent variable with respect to the in-dependent variable. In each graphic the feature quality is seen as favourable (for classification) if the genre (coloured lines) are clearly separable and unfavourable if the genres intersect frequently as this would mean that the feature descriptor produces the same value for more than one genre indicating similarity between genres whereas the goal of these descriptor are to distinguish between genres. Spectral centroid is seen as a very good metal and disco descriptor among other genres.

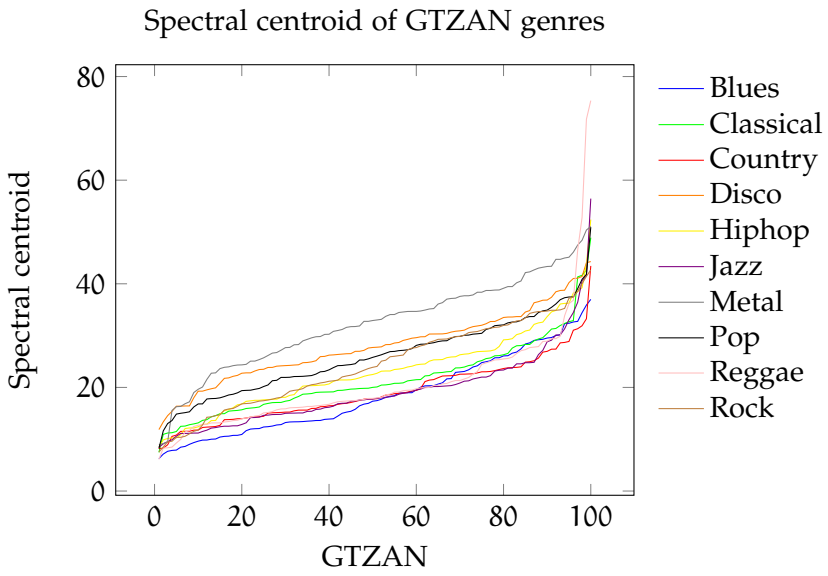


Figure 19: Spectral centroid feature values for 10 GTZAN genres using the mean representation.

Representation	KNN	MP	NB	RF	LLRM	SVM
Mean	24.80%	32.60%	28.10%	27.90%	30.00%	28.60%
MFCC	26.40%	27.80%	30.70%	28.80%	32.10%	28.30%
20-bin FH	24.80%	26.10%	23.70%	25.60%	25.50%	20.70%
Area Moments	18.30%	19.60%	17.50%	18.80%	17.60%	13.20%

Table 13: Classification scores for different feature representations for spectral centroid using a variety of classification techniques.

Table 13 shows the results of using spectral centroid to classify 10 GTZAN genres using multiple classification techniques with 10-fold cross validation. Although spectral centroid is best represented by MFCCs for this problem, it is seen that the mean representation classified with a multilayer perceptron yield the optimal result by 0.50% compared to using linear logistic regression models. Table 14 shows the results of taking the strongest frequency of spectral centroid (in Hz), not much difference is noted in this case. The definition of spectral centroid is now extended to define other shape statistics used for signal classification.

Representation	KNN	MP	NB	RF	LLRM	SVM
Mean	24.80%	32.60%	27.90%	27.90%	30.00%	15.30%
MFCC	26.40%	27.80%	30.30%	30.00%	32.20%	28.30%
20-bin FH	24.80%	26.10%	23.70%	25.60%	25.50%	20.70%
Area Moments	18.20%	18.20%	17.60%	20.40%	17.80%	12.40%

Table 14: Classification scores for different feature representations for strongest frequency of spectral centroid using a variety of classification techniques.

6.3.3.2 *Spread, Kurtosis and Skewness*

Extending the definition of spectral centroid in Equation 6.14 as

$$[\mathbb{SC}]_i = \frac{\sum_{n=0}^{N-1} f^i(n)x(n)}{\sum_{n=0}^{N-1} x(n)}, \tag{21}$$

where $[\mathbb{SC}]_i$ represents the spectral centroid at position i. $[\mathbb{SC}]_i$ is the normalised i^{th} moment of the magnitude about zero. $[\mathbb{SC}]_i$ allows us to define spread, kurtosis and skewness as:

Definition 6.15. *Spread*
The spectral spread of a signal is given as

$$[\mathbb{S}]_{spread} = \sqrt{[\mathbb{SC}]_2 - [\mathbb{SC}]_1^2}; \tag{22}$$

Definition 6.16. *Kurtosis*
The spectral kurtosis of a signal is given as

$$[\mathbb{S}]_{kurtosis} = \frac{-3[\mathbb{SC}]_1^4 + 6[\mathbb{SC}]_1[\mathbb{SC}]_2 + 4[\mathbb{SC}]_1[\mathbb{SC}]_3 + [\mathbb{SC}]_4}{[\mathbb{S}]_{spread}^4} - 3; \tag{23}$$

Definition 6.17. *Skewness*
The spectral skewness of a signal is given as

$$[\mathbb{S}]_{skewness} = \frac{2[\mathbb{SC}]_1^4 + 3[\mathbb{SC}]_1[\mathbb{SC}]_2 + [\mathbb{SC}]_3}{[\mathbb{SC}]_{spread}^3}. \tag{24}$$

Using linear logistic regression models to classify GTZAN genres yields 36.2% using only shape statistics. In the next section we introduce spectral rolloff which measures how much of the power spectrum is right-skewed.

6.3.4 *Spectral Rolloff*

According to Peeters [2004]; Bergstra *et al.* [2006], spectral rolloff point is the frequency so that 85% of the signal energy is contained below this frequency. It is correlated to the harmonic/noise cutting frequency [Peeters 2004]. Spectral rolloff has been used for music genre classification [Tzanetakis and Cook 2002; Alexandre-Cortizo *et al.* 2005; Li *et al.* 2003] and speech classification [Alexandre-Cortizo *et al.* 2005; Scheirer and Slaney 1997]. The spectral rolloff is measured by the following equation:

$$\sum_{0}^{fc} a^2(f) = 0.85 \sum_{0}^{sr/2} a^2(f), \tag{25}$$

where f_c is the spectral roll-off frequency, a is the amplitude of the frequency and $\frac{sr}{2}$ is the Nyquist frequency. The Nyquist frequency is defined as:

Definition 6.18. *Nyquist frequency*
The highest frequency that can be represented in a DTS of a specified sampling frequency. Nyquist frequency is equivalent to $\frac{1}{2}$ of the sampling rate.

As seen in Figure 20, spectral rolloff classifies metal, hiphop, and disco very well. This capable classification - using mean representation - is maintained in Table 15, however, using random forests with a 20 bin feature histogram yields the best classification score in this setting.

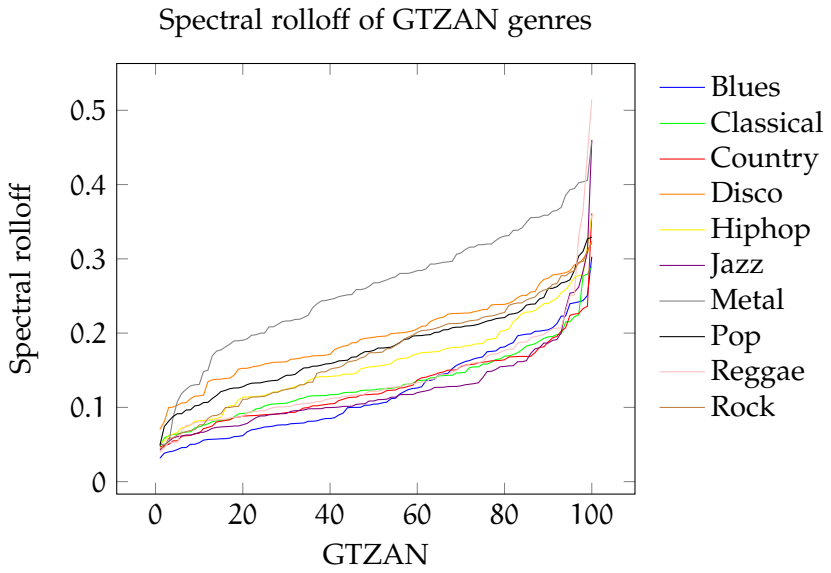


Figure 20: Spectral rolloff feature values for 10 GTZAN genres using the mean representation.

Now that we have explored a number of magnitude-based features we can summarise this information to effectively classify GTZAN genres.

6.4 CONCLUSION AND DISCUSSION

In Section 6.2 and Section 6.3 we explored magnitude-based features for music genre classification using the GTZAN dataset. Some features produced exceptional results

Representation	KNN	MP	NB	RF	LLRM	SVM
Mean	26.80%	35.90%	30.30%	29.10%	33.60%	27.60%
MFCC	26.70%	30.20%	30.70%	28.00%	31.90%	28.70%
20-bin FH	28.10%	29.20%	29.00%	33.80%	33.00%	23.10%
Area Moments	19.80%	21.10%	20.30%	21.50%	20.40%	18.70%

Table 15: Classification scores for different feature representations for spectral rolloff using a variety of classification techniques.

(MFCCs, Compactness, loudness) while other features did not distinguish genre correctly and added unnecessary dimensionality to our design matrix. In this section we summarize the contributions of Section 6.2 and Section 6.3 into a design matrix that best describes a music genre database (GTZAN). Optimistically, this research argues that the GTZAN is a good representation of other music databases (such as MusicBrainz), and the ideas expressed in this research can be applied to larger music datasets.

Feature list with representation (149)	
Slope (1)	Mean
Compactness (2)	Mean
Decrease (1)	Mean
Loudness (26)	Mean
Onset Detection (1)	Mean
Octave Based Signal Intensity (17)	Mean
Peak-based features (4)	Mean
Spectral Flux (4)	MFCC
Spectral Variability (4)	MFCC
MFCC (52)	MFCC
Flatness (20)	Mean
Shape Statistics (11)	Mean/MFCC
Spectral Rolloff (2)	Mean
Peak Flux (2)	20-bin FH
Crest Factor (10)	Mean
Strongest Freq of FFT Max (4)	MFCC

Table 16: A list of magnitude-based feature for genre classification.

Table 16 presents a list of features selected from Section 6.2 and Section 6.3, the feature dimensionality[9] is given in parenthesis. The feature selection is as follows: spectral slope is represented by the mean; compactness also given by the mean and standard deviation (2); spectral decrease is given by the mean (1); loudness is given by 24 coefficients all of which are represented by the mean (24), plus perceptual sharp-

9 The number of coefficients that represent the feature itself, and hence the number of dimensions that the feature uses.

ness and spread given by the mean (+2); onset detection is given by the mean (1); octave based spectral intensity is given by two components - OBSI (9 components) and OBSIR (8 components), all represented by the mean; peak-based features given by the mean and standard deviation, these peak-based features include peak centroid (2) and peak smoothness (2); spectral flux represented by 4 MFCC coefficients (4); spectral variability also represented by 4 MFCC coefficients (4); Mel-frequency Cepstral coefficients given in the MFCC representation ($13 \times 4 = 52$); spectral flatness, given by the mean of 19 bands + the overall average (20); shape statistics that include centroid and strongest frequency of centroid represented by MFCCs (4 + 4), and spread, kurtosis & skewness represented by mean (3); spectral rolloff, given by average and standard deviation (2); peak flux represented by a 20-bin feature histogram; crest factor represented by the mean; and finally, the strongest frequency of FFT represented by MFCC (4). Altogether there are 149 features that make up the design matrix.

Using all of the selected features in Table 16 with the corresponding representation we achieve 77.8% successful classification using linear logistic regression models; 79.5% using multilayer perceptron; 62.7% using random forests; 62.6% using naïve Bayes; 70.6% using k-nearest neighbours; and 22% using support vector machines. In this section content-based features derived from the magnitude spectrum are presented to demonstrate the effectiveness of using signal transforms for music genre classification. Altogether this section organised a design matrix of 149 dimensions + 1 for the genre label and even though reducing data dimensionality could have resulted in more precise classification - ideas are explored in Chapter 10 - the main objective of this chapter is set the stage for more families of features to be introduced.

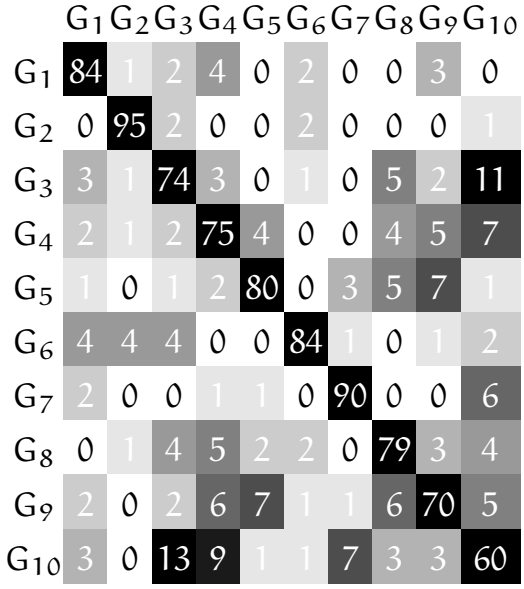


Figure 21: Multilayer perceptron classification (79.5%) of GTZAN genres using only magnitude-based features.

Figure 21 shows the confusion matrix for 10 GTZAN genres using only magnitude-based features with a multilayer perceptron. The row and column labels represent genre labels where: G_1 = Blues, G_2 = Classical, G_3 = Country, G_4 = Disco, G_5 = Hiphop, G_6 = Jazz, G_7 = Metal, G_8 = Pop, G_9 = Raggae, and G_{10} = Rock. To demonstrate the fuzziness between genres, as explained in Section 1.2, it is noted that in Figure 21 there are 13 rock pieces which were classified as country and 11 country pieces classified as rock.

Example 6.19. Another example demonstrates the fuzziness between rock and disco music where 9 rock pieces where classified as disco and 7 disco pieces were classified as rock.

Another pressing factor is the amount of time taken by the multilayer perceptron to build a classification model. While the other classifiers took between 0.06-10.2 seconds to build a model, the multilayer perceptron took 107.93 seconds. Although the multilayer perceptron produced an outstanding result in classification, this time constraint makes us question the reliability of the model to handle more features or truly enormous datasets, much like MusicBrainz[10].

In the next section we introduce another family of features derived from the root mean square of the DTS. This feature opens doors to analyse beats within a melody offering a constructive way to perform rhythm analysis.

10 MusicBrainz having 16 000 000 music files.

7

TEMPO DETECTION

Most music display regular rhythmic formation that creates an impression of tempo. With the purpose of understanding the nature of music to perform genre classification, tempo must be understood and preserved as a feature description. In this chapter we establish tempo detection schemes for music genre classification. Having already established a method to detect the vitality in a music excerpt by using spectral energy, which is the root mean square of the music signal, we present in this chapter the Beat Histogram as a crucial design matrix. The Beat Histogram is then extended to craft more descriptive and useful features regarding tempo.

7.1 INTRODUCTION

Every piece of music contains some rendition of rhythm, and thus every piece of music contains at least one beat, where a beat is defined as one rhythmic unit. Melodies, chordal progressions, heterophony, and scales all rely on rhythmic pattern designs to guide and direct successful and structured music. In music, rhythm is understood by the use of time signatures which expresses an intended relationship between estimated duration and actual time.

The main beats of a piece can be expressed as the point where the listener would clap her hands while listening to a piece of music. These beats can be extracted and measured to provide information about every beat in a given threshold, together these beats yield the beat histogram. A problem arises where multiple music pieces will have very similar rhythmic structure belonging to different genres. In this case our analysis needs to define beat strength, which will help us distinguish pieces of music with the same rhythm from different genres.

Example 7.1. As an example, understanding beat strength and other features derived from the beat histogram, can help us distinguish a rock song, with a higher beat strength, from a classical piece at the same tempo.

Energy is firstly introduced, being a fundamental concept for studying rhythm, along with some variations of energy that include the relative difference function and fractions of low energy; and finally, we introduce the beat histogram and beat strength features.

7.2 ENERGY

Energy is a fundamental descriptor used in speech and audio processing [Lu *et al.* 2002]. The *root mean square* is a central concept when understanding energy and is defined in Section A.1. Energy is defined as follows:

Definition 7.2. *Energy*
Energy, denoted E_s, is measured by calculating the RMS of a DTS. E_s is defined as:

$$E_s = \sqrt{\frac{\sum_{i=0}^{N-1} x_i^2}{N}}. \tag{26}$$

Figure 22 shows the energy feature values of 10 GTZAN genres with one hundred 30 sec excerpts for each genre. The figure shows the range of values for energy for each genre represented by the mean. In the figure classical, jazz, pop, and hiphop genres have distinguishing energy ranges compared to other genres.

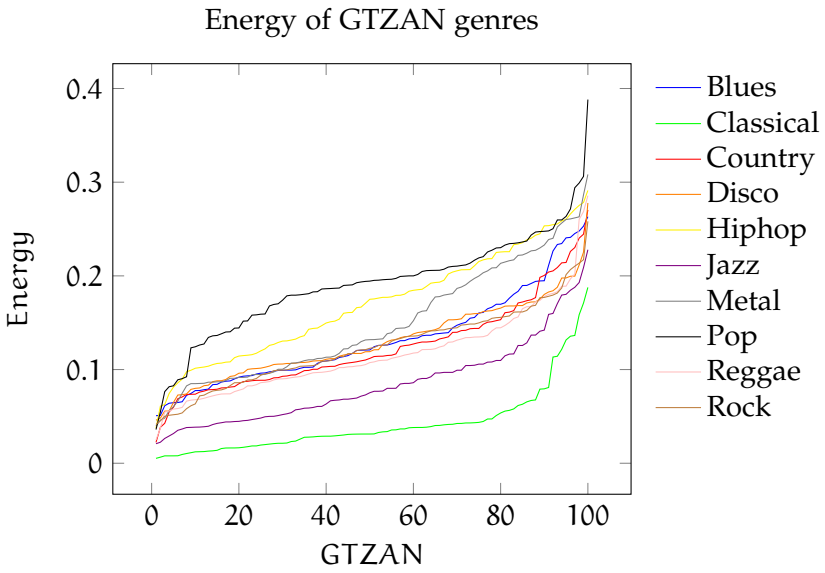


Figure 22: Energy feature values for 10 GTZAN genres using the mean representation.

According to Table 17, using energy represented by MFCCs we achieve accuracies of 40.3% using naïve Bayes; 40.8% using support vector machines; 37.7% using the multilayer perceptron; 36.3% using linear logistic regression models; 33.7% using k-nearest neighbours; and 39% using random forests with 10-fold cross validation. These results show that energy is a capable feature for music genre classification using GTZAN genres, although, genres such as blues, disco, country, and rock display very similar energy values and so other adaptations of energy can also be investigated.

If we examine the arithmetic average of the first n windows of a signal (for our experiments we took $n = 100$) and calculate to what fraction of these values are below the average, then we can calculate the percentage of silence that exists in the

Representation	KNN	MP	NB	RF	LLRM	SVM
Mean	28.90%	35.70%	27.50%	33.90%	32.60%	20.80%
MFCC	33.70%	37.70%	40.30%	39.00%	36.30%	40.80%
20-bin FH	25.90%	22.40%	21.60%	24.80%	20.70%	17.90%
Area Moments	25.00%	24.50%	26.20%	28.00%	25.20%	24.10%

Table 17: Classification scores for different feature representations for energy using a variety of classification techniques.

signal - let us call this feature the *fraction of low energy*. Onset detection can also be achieved through the study of energy.

Every note in a piece of music consists of an onset[1], and could be a useful tool for music genre detection. Taking the log of the derivative of energy tells us something about the increase, change in distribution and pitch of spectral energy, this feature is commonly referred to as the *relative difference function* of a DTS. The relative difference function represented by MFCCs achieve accuracies of 21.2% using naïve Bayes; 21.7% using support vector machines; 20.7% using the multilayer perceptron; 20.7% using linear logistic regression models; 19.2% using k-nearest neighbours; and 23.7% using random forests with 10-fold cross validation on GTZAN genres.

The study of energy has allows us to examine tempo, being a fundamental aspect of music. The next section employs a commonly used tempo detection scheme, using energy as a primary component, called the beat histogram.

7.2.1 *Beat Histogram*

The beat histogram is an arrangement of signal strength to yield rhythmic intervals. This is accomplished by measuring the RMS energy of n consecutive windows and taking the fast Fourier transform of the result. This type of feature will produce a very large design matrix and so a simple feature representation is needed. In our experiments the mean feature representation outperformed MFCC and the 20-bin feature histogram. Using the beat histogram represented by the mean we achieve accuracies of 21.4% using naïve Bayes; 34.1% using support vector machines; 33.1% using linear logistic regression models; 29.2% using k-nearest neighbours; and 29.7% using random forests with 10-fold cross validation.

Exploiting properties of the beat histogram can prove beneficial, for example taking the sum of all values in the beat histogram gives us a feature that shows the significance of regular beats in a music piece. Table 18 shows the sum of all beats of GTZAN genres using multiple classification techniques and feature representations. The MFCC representation seems to on average outperform all of the other representations and the multilayer perceptron, even though takes much longer to build than

1 Onset involves the rise in amplitude when a sound occurs. This rise in energy begins at zero to some amplitude peak.

Representation	KNN	MP	NB	RF	LLRM	SVM
Mean	26.10%	27.70%	21.50%	27.90%	24.60%	21.20%
MFCC	25.80%	28.50%	28.10%	29.60%	27.40%	23.50%
20-bin FH	17.90%	16.60%	17.30%	19.60%	19.40%	14.90%
Area Moments	26.90%	24.90%	21.50%	26.60%	23.50%	13.40%

Table 18: Classification scores for different feature representations for beat sum using a variety of classification techniques.

other classification techniques, appears to classify genre better than any of the other techniques.

Another way to exploit the properties of the beat histogram is giving the value of the highest bin of the beat histogram as the strongest beat in the signal. Figure 23 shows the strongest beat of each excerpt of GTZAN genres. It is noted that blues and pop share similar strongest beats, whereas hiphop, reggae, and disco do not.

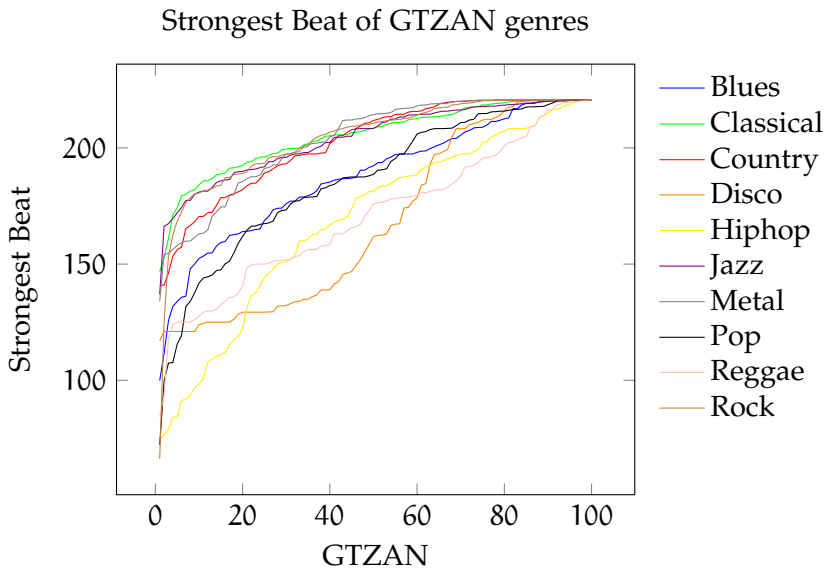


Figure 23: Strongest beat feature values for 10 GTZAN genres using the mean representation.

Table 19 shows the strongest beat of GTZAN genres using multiple classification techniques and multiple feature representations. The MFCC representation seems to outperform all of the other representations and K-nearest neighbours classification appears to classify genre better than any of the other techniques.

Representation	KNN	MP	NB	RF	LLRM	SVM
Mean	22.60%	21.50%	18.80%	22.30%	19.80%	20.10%
MFCC	20.10%	21.70%	18.80%	21.10%	18.70%	19.30%
20-bin FH	18.60%	16.20%	14.60%	19.90%	14.50%	14.90%
Area Moments	21.30%	20.50%	17.60%	21.70%	14.30%	16.80%

Table 19: Classification scores for different feature representations for strongest beat using a variety of classification techniques.

We can further use the strongest beat feature to calculate the strength of the strongest beat, which will measure to what ratio is the strongest beat stronger compared to other beats. In the next section the ideas of this chapter are summarised and the contributions of this section are given in terms of a design matrix.

7.3 CONCLUSION AND DISCUSSION

Tempo is a mandatory aspect of music genre that is only interpreted in a few different ways. For example, a $\frac{4}{4}$ time signature can be used in almost every type of genre. Therefore, we can expect that identifying tempo in a music signal is not a sufficient feature on its own to detect genre.

Example 7.3. A typical example would be using the waltz rhythmic pattern which can be found in jazz (e.g. *Mississippi Waltz* by the Memphis Jug Band); classical (e.g. *Invitation to the Dance* by Carl Maria von Weber); pop (e.g. *Kiss from a Rose* by Seal); and rock (e.g. *Breakaway* by Kelly Clarkson).

Thus, we will not expect to get extraordinary results by classifying music genre by rhythmic features alone, however, the contributions of rhythmic features when combined with other features are worthwhile.

Feature list with representation (362)	
Energy (2)	Mean
Fraction of low energy (2)	Mean
Beat Histogram (342)	Mean
Strongest Beat (2)	Mean
Strength of the Strongest Beat (2)	Mean
Beat Sum (4)	MFCC
Relative Difference Function (4)	MFCC
Temporal Shape Statistics (4)	Mean

Table 20: Tempo-based feature list.

Table 20 presents a list of rhythmic features with the feature dimensionality[2] given in parenthesis. The feature selection with each feature's representation is given as follows: energy is represented by the mean and standard deviation (1); fraction of low energy is given by the mean and standard deviation (2); the beat histogram is given by the mean and standard deviation (342), along with the strongest beat (2) given by the mean, and strength of the strongest beat (2) given by the mean; the beat sum given by MFCC representation (4); the relative difference function given by MFCC values; and finally temporal statistics given by the mean alone (4). Altogether there are 362 features, + 1 for the genre label, that make up the design matrix.

2 The number of coefficients that represent the feature itself, and hence the number of dimensions that the feature uses.

Using all of the selected features in Table 20 with the corresponding representation we achieve accuracies of 49.3% successful classification using linear logistic regression models; 46.5% using multilayer perceptron; 37.4% using random forests; 23.3% using naïve Bayes; 36.1% using k-nearest neighbours; and 36.5% using support vector machines.

Figure 24 shows the confusion matrix for 10 GTZAN genres using only tempo-based features with linear logistic regression models. The row and column labels represent genre labels where: G_1 = Blues, G_2 = Classical, G_3 = Country, G_4 = Disco, G_5 = Hiphop, G_6 = Jazz, G_7 = Metal, G_8 = Pop, G_9 = Reggae, and G_{10} = Rock. To demonstrate a major drawback of using only tempo-based features for music genre classification we can observe how many music pieces are misclassified as blues. In this case, since many blue music excerpts in the GTZAN dataset contain common rhythmic patterns, many music pieces - although from other genres - are classified as blue. However, it is hypothesised that using temporal feature with timbre and pitch features will clear this obscurity.

	G_1	G_2	G_3	G_4	G_5	G_6	G_7	G_8	G_9	G_{10}
G_1	28	1	3	13	9	11	10	13	5	7
G_2	0	84	3	0	0	9	2	0	0	2
G_3	1	3	34	8	0	19	14	3	3	15
G_4	11	0	3	45	8	4	4	6	12	7
G_5	4	0	1	14	52	2	0	15	9	3
G_6	8	14	15	1	2	39	2	3	7	9
G_7	1	0	7	6	1	4	65	5	0	11
G_8	1	1	5	6	13	1	5	55	3	10
G_9	3	2	3	10	9	6	3	3	61	0
G_{10}	12	5	14	5	2	12	13	4	3	30

Figure 24: A tempo-based design matrix using linear logistic regression models classifier achieving 49.3% accuracy.

In the next section we introduce features that are commonly used in signal processing for speech and pitch detection. These are most likely good features, for genre classification, as tone and singing are very common aspects in musical content. This research will employ the most effective signal processing features for speech recognition and observe their effects on automatic music genre classification.

8

PITCH AND SPEECH DETECTION

Zero-crossing rate is frequently used as a feature for signal classification as a thorough percussive descriptor. In this chapter we explore tonal and speech-based features for music genre classification. We show how a signal fluctuations and zero crossings can be used to detect some aspects of pitch. Using only pitch and speech based features we obtain 67.3% classification accuracy using multilayer perceptions on GTZAN genres.

8.1 INTRODUCTION

Pitch is a perceived characteristic contained in the frequency of music content. Most music of the same genre exhibit melodies that are just combined notes from a scale set[1]. For example, most notes from an impressionistic piece are taken from whole-tone scales, whereas notes from a jazz pieces of music are taken from pentatonic scales. However, often environmental sounds overtone pitch, disguising available pitch-related elements, which make it difficult to extract pitch computationally. Even human auditory systems can find it difficult to distinguish pitch under these conditions. Therefore, some sort of pitch extraction mechanisms need to be adopted to retrieve these pitch elements though the environmental sounds. In this chapter we explore pitch and speech related algorithms as an amalgam of these characteristics are hypothesised to describe singing. Together, pitch and speech detection schemes can help us understand gliding; portamento; or even vibrato (amplitude modulation).

8.2 PITCH DETECTION

Two important pitch-based features are discussed in this chapter: *zero crossing rate*, a primeval pitch detector; and *amplitude modulation*: a description of tone that describes how pitch modulates over time.

8.2.1 *Amplitude Modulation*

For many musical instruments amplitude periodic modulation[2] is a distinctive quality. Figure 25 shows the DTS of a note played by a saxophone[3]. In Figure 25 there

1 A scale is simply a set of notes.
2 The way the signal oscillates or appears to oscillate over time.
3 The saxophone belongs to the family of wind instruments.

appears to be a periodic waveform with a period of about 80 time units. The different periods may not be exactly the same as every other period, but this amplitude modulation or "fluttering" can be quite descriptive for instrument recognition [Eronen and Antti 2001].

Amplitude modulation of a note played by a saxophone

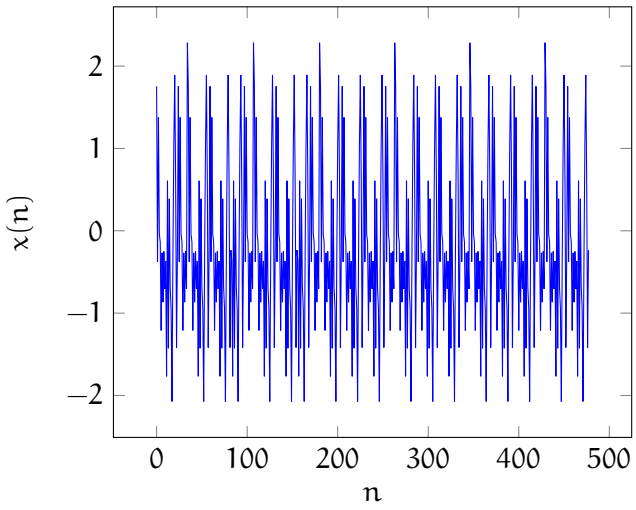


Figure 25: Amplitude modulation of a DTS.

According to Eronen and Antti [2001], style introduces characteristic amplitude variation into music tones. Iverson and Krumhansl [1993] observed that changing amplitude envelopes leads to similarity judgments on musical timbre. Eronen and Antti [2001] maintains that the energy (Equation 7.2) envelope is useful to extract features measuring amplitude modulation (AM). Martin [1999] observed that heuristic strength and frequency of AM can be calculated at two frequency ranges. Eronen and Antti [2001] stated that the first range is between 4 and 8 Hz (where the AM is in conjunction with vibrato) and the second range is between 10 to 40 Hz which correspond to "graininess" or "roughness" of the tone. In order to calculate the AM of a DTS the following steps are followed for each of the described ranges [Mathieu *et al.* 2010]:

1. **Step 1:** Find the frequency of maximum energy in range.
2. **Step 2:** Calculate the difference of the energy of this frequency and the mean energy over all frequencies.
3. **Step 3:** Calculate the difference of the energy of this frequency and the mean energy in range.
4. **Step 4:** Take a product of the above two values.

8.2.2 *Zero Crossing Rate*

The *zero crossing rate* (ZCR) is the frequency of sign changes that occur along a discrete-time signal [Chen 1988]. Being a thorough percussive descriptor[4], this feature has been used in both speech recognition as well as in audio information retrieval [Gouyou *et al.* 2000]. The use of this feature was extended for classification of percussive sounds [Gouyon *et al.* 2000]; modulation classification [Hsue and Soliman 1990]; and music genre classification [Xu *et al.* 2003]. An indicator function is firstly introduced and is then extended to define the ZCR of a DTS.

Definition 8.1. *The Indicator Function*
If $A \subseteq X$, Then

$$\mathbb{1}_A : X \to \{0, 1\}, \tag{27}$$

is an indicator function defined as:

$$\mathbb{1}_A(x) := \begin{cases} 1 \text{ IF } x \in A \\ 0 \text{ IF } x \notin A \end{cases}. \tag{28}$$

Definition 8.2. *Zero Crossing Rate*
A formal definition of the ZCR is given by:

$$\text{ZCR} = \frac{1}{T-1} \sum_{t=1}^{T-1} \mathbb{1}\{x_t x_{t-1} < 0\}, \tag{29}$$

where x is the discrete-time signal of length T. The indicator function, $\mathbb{1}\{x_t x_{t-1} < 0\}$, is 1 if the argument $\{x_t x_{t-1} < 0\}$ is true or 0 if the argument is false. If one is to encounter a *monophonic*[5] discrete-time signal, then the ZCR can be used as a primeval pitch detector.

Figure 26 shows the ZCR values of 10 GTZAN genres with one hundred 30 sec excerpts for each genre. The figure shows the range of values of ZCR for each genre represented by the mean. In the figure metal, disco, and hiphop genres have distinguishing energy ranges compared to other genres. According to Table 17, using ZCR represented by the mean we achieve accuracies of 32.3% using naïve Bayes; 28.1% using support vector machines; 32.50% using the multilayer perceptron; 33% using linear logistic regression models; 29.1% using k-nearest neighbours; and 29.6% using random forests with 10-fold cross validation. If we calculate the frequency of zero crossings in a sample of a discrete time signal we can identify the most intense component of that sample which could have potential for genre classification. This component is commonly referred to as the strongest frequency using ZCR and is measured in Hz.

4 Able to distinguish between musical instruments that are sounded by begin struck or scraped by a beater, by hand, or by a similar instrument.
5 Melody without accompanying harmony.

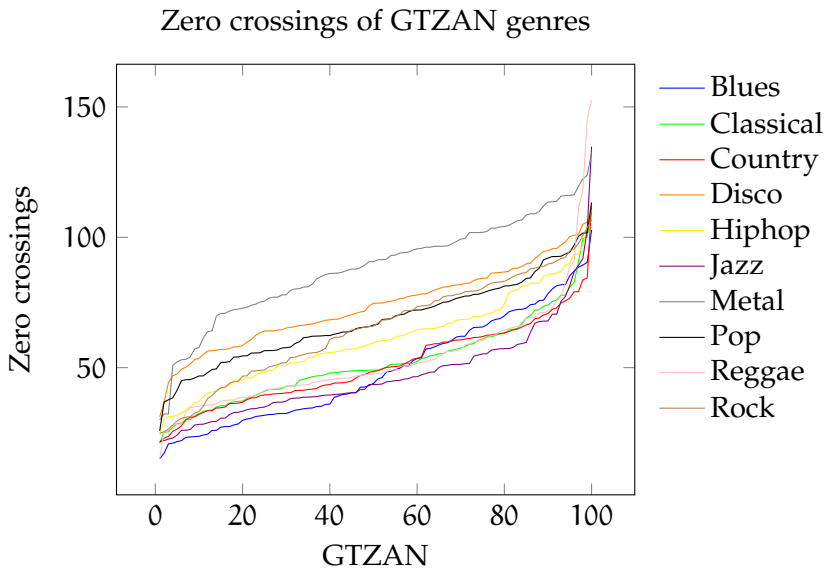


Figure 26: Zero crossings feature values for 10 GTZAN genres using the mean representation.

8.3 SPEECH DETECTION

Speech detection is an ever growing field in computer and information science. In this section we attempt to use speech-based features for music genre classification. Using one of the most powerful speech analysis techniques, linear prediction coefficients, we achieve exceptional results for genre classification. We begin by defining autocorrelation coefficients as a feature for genre detection.

8.3.1 *Autocorrelation Coefficients*

Auto-correlation is a tool used to define features such as linear predictor coefficients (LPC). Auto-correlation itself is also considered as a standard feature and will be used as such in this research. The definition of auto-correlation is aided by the following definitions: integral transform; complex conjugates (Section A.7); and lastly, the discrete-time autocorrelation coefficient.

Definition 8.3. *Integral Transform*
An integral transform is defined by the following transformation of a function f:

$$(Tf)(x) = \int_{t_1}^{t_2} K(t,x)f(t)dt, \tag{30}$$

where f is the transform function and K is a specified choice for the transformation operator.

Definition 8.4. *Discrete-Time Auto-correlation Coefficient*
When the autocorrelation function is normalised by mean and variance, it is sometimes referred to as the autocorrelation coefficient [Dunn 2010]. Given a signal $f(t)$,

the continuous autocorrelation $R_{ff}(\tau)$ is most often defined as the continuous cross-correlation integral of $f(t)$ with itself, at lag τ:

$$R_{ff}(\tau) = (f(t) * \bar{f}(-t))(\tau) \tag{31}$$

$$= \int_{-\infty}^{\infty} f(t+\tau)\bar{f}(t)dt \tag{32}$$

$$= \int_{-\infty}^{\infty} f(t)\bar{f}(t-\tau)dt, \tag{33}$$

where $\bar{f}$ represents the complex conjugatex (Section A.7) and * represents convolution (Section A.6). For a real function, $\bar{f} = f$.

Makhoul [1975] presents a compact way to produce linear prediction coefficients using Levinso-Dubin algorithm that make use of autocorrelation coefficients. Li *et al.* [2001] used this algorithm to perform automatic music genre classification. An adaptation of this algorithm will be used in this research for classifying GTZAN genres. The adaptation will further use LPC to define *line spectral frequency* (LSF) using the methods by Bäckström and Magi [2006]; Schussler [1976].

8.4 ENVELOPE SHAPE STATISTICS

Expanding on Section 6.3.3, which provide spectral shape statistics on the magnitude spectrum, we can similarly calculate the spectral statistics of each envelope[6] of the DTS. These provide ideal pitch descriptors for tonal features. In the next section all of the ideas in this chapter are summarised and a design matrix is given that describe pitch and speech based features.

8.5 CONCLUSION AND DISCUSSION

In this section we summarize the contributions of the previous sections and introduce a design matrix that best classifies a music genre database (GTZAN). Table 21 presents a list of pitch and speech based features, the feature dimensionality is given in parenthesis. The feature selection is as follows: 49 autocorrelation coefficients represented by the mean; 8 amplitude modulation coefficients represented by the mean; zero crossing represented by MFCCs; 4 envelope statistics represented by the mean (4); and finally 8 LSF coefficients with 2 linear predictor coefficients. Altogether there are 75 features (+ 1 for the genre label) that make up this design matrix.

Using all of the selected features in Table 21 with the corresponding representation we achieve accuracies of 79.5% successful classification using linear logistic regression models; 78.9% using multilayer perceptron; 65.8% using random forests; 65.3% using naïve Bayes; 71.7% using k-nearest neighbours; and 20.6% using support vector machines.

6 An amplitude envelope given by Hilbert's transform, low-pass filtering and decimation

Feature list with representation (75)	
Autocorrelation Coefficients (49)	Mean
Amplitude Modulation (8)	Mean
Zero Crossing (4)	MFCC
Envelope Statistics (4)	Mean
LFS (10)	Mean

Table 21: A list of pitch and speech based features for genre classification.

Figure 27 shows the confusion matrix for 10 GTZAN genres using only pitch and speech based features with a multilayer perceptron. The row and column labels represent genre labels where: G_1 = Blues, G_2 = Classical, G_3 = Country, G_4 = Disco, G_5 = Hiphop, G_6 = Jazz, G_7 = Metal, G_8 = Pop, G_9 = Reggae, and G_{10} = Rock.

In the next section we introduce a core feature for music genre detection derived from the ConstantQ transform. This feature not only serves as a strong classifier on its own, but can also can be used for identifying chordal structures in music content.

	G_1	G_2	G_3	G_4	G_5	G_6	G_7	G_8	G_9	G_{10}
G_1	67	1	4	2	4	6	9	0	1	6
G_2	0	90	4	0	0	6	0	0	0	0
G_3	8	3	52	3	0	16	3	2	2	11
G_4	2	1	1	73	4	1	3	4	4	7
G_5	5	0	1	6	52	0	6	10	19	1
G_6	4	9	9	2	0	63	0	3	4	6
G_7	2	0	1	2	3	0	85	0	0	7
G_8	0	0	4	2	7	2	0	75	2	5
G_9	2	0	5	5	8	1	1	4	67	7
G_{10}	9	0	12	6	3	4	8	7	5	46

Figure 27: Multilayer perceptron classification (67.3%) of GTZAN genres using pitch and speech based features.

9 CHORDAL PROGRESSIONS

Introducing spectral feature extraction to genre detection problems created opportunities to exploit single characteristics of music. Chord structure and progressions has been a defining trait of music for many years but had gone unnoticed in recent music genre detection schemes. In this chapter we revisit chroma as a viable timbre feature [Aucouturier and Pachet 2002] and use a MFCC representation technique to classify a benchmark dataset. Furthermore, we show that MFCC representation provides better classification than using the arithmetic mean representation on chroma. Using MFCC-based chroma and mean-based MFCCs we achieve 55.2% precision on the GTZAN dataset using linear logistic regression models.

9.1 INTRODUCTION

Since the early development of spectral genre classification, timbre features have been used to search discrete music signals in an attempt to describe some aspects of music [Mandel and Ellis 2005].

Timbre features describe content based musical aspects such as instrumentation, rhythm and pitch modulation. Ellis [2007] reviewed the importance of music classification and used chroma and MFCCs to identify artists.

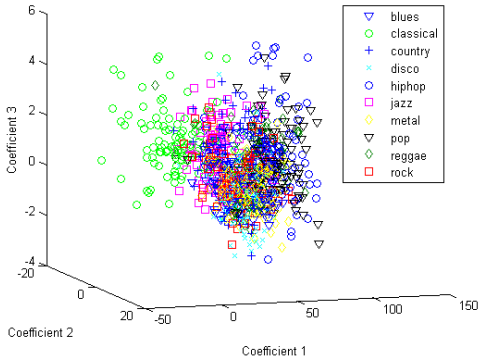


Figure 28: Global distribution of MFCC-based chroma (3 coefficients) on GTZAN genres.

Chroma has been identified as a timbre feature used to describe chord structure and is barely influenced by instrumentation and tempo [Ellis and Poliner 2007]. According to Ellis [2007], chroma is commonly used to match "cover songs"[1]. Since cover music contains more-or-less the same harmonic content but with altered instrumentation and rhythm, it is safe to assume that chroma can be used to identify music based on little influence from rhythm and instrumentation.

1 A new performance or recording of a previously performed recording by someone other than the original artist.

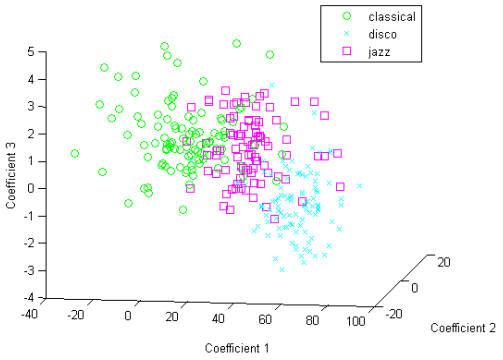


Figure 29: Global distribution of MFCC-based chroma (3 coefficients) on GTZAN's classical, disco, and jazz genres.

Chroma is defined as a 12 component design matrix where each dimension represents the intensity associated with a particular semitone, regardless of octave [Ellis 2007]. This chapter implements MFCC-based chroma by extracting MFCCs derived from the 12 component chroma design matrix. Comparing the classification results of regular chroma verses MFCC-derived chroma, we achieve a 13.8% increase in classification precision using naïve Bayes. Figure 28 shows a visual model of the first 3 dimensions of MFCC-based chroma applied to the GTZAN dataset.

Since the components of chroma describe the distribution of semitones in a piece of music, it also tells us how notes are arranged and thus tells us information about chordal harmonies. Therefore, modelling chroma can tell us if a particular genre displays an attachment or relation to harmonic chordal progressions, as some genres do.

Example 9.1. Pop and rock music might have the same chord progressions while jazz has more distinct/unusual chord progression compared to other genres.

Feature	Classifier	Accuracy	Time (sec)
Chroma (mean)	Multilayer Perception	28.60%	2.340
	Naïve Bayes	24.20%	0.010
	Support Vector Machine	16.00%	0.660
	Linear Logistic Regression Models	31.20%	2.220
	Random Forest	30.80%	0.160
Chroma (MFCC)	Multilayer Perception	36.50%	9.470
	Naïve Bayes	38.00%	0.001
	Support Vector Machines	35.90%	0.970
	Linear Logistic Regression Models	36.60%	2.670
	Random Forest	37.20%	0.200

Table 22: Classification scores for chroma on GTZAN genres.

Figure 29 shows the global distribution of MFCC-based chroma (3 coefficients) on GTZAN's classical, disco, and jazz genres. We can see that although these genres can exhibit variant tempo and instrument patterns, they are still clustered closely in the figure. This is because of their chordal sounds that MFCC-chroma detects. In order to fully understand the potential of MFCC-based chroma three experiments are conducted. Firstly, we examine the classification of chroma alone using the MFCC and mean representation; secondly, we observe mean-based MFCC classification; and lastly, we observe the different precision achieved from mean and MFCC -based chroma with the mean-based MFCC feature.

9.1.1 *Results*

In Table 22 we used MFCC-based chroma and mean-based chroma features alone to classify the GTZAN database genres. By modelling chroma features alone we are able to see the global distribution of chroma on each excerpt in the database and thus are able to compare feature extraction strategies.

Feature	Classifier	Accuracy	Time to build model
MFCC (mean)	Multilayer Perception	47.20%	2.180
	Naïve Bayes	45.00%	0.001
	Support Vector Machine	46.70%	0.700
	Linear Logistic Regression Models	46.70%	1.120
	Random Forest	46.10%	0.110

Table 23: Classification scores for MFCCs on GTZAN genres.

From Table 22 we find that MFCC-based chroma outperforms mean-based chroma for all of the classification techniques. Table 23 shows the classification of mean-based MFCCs on all of the GTZAN dataset genres. Although the multilayer perceptron takes the longest to build, it performed a little better compared to all of the other classification techniques. We can see that regular MFCCs only produce about 5% better classification compared to MFCC-based chroma. However, we hypothesise that since chroma yields information about chordal progression and MFCCs describes linguistic content, then we should expect much better results when these two features are combined. The concepts presented here not only show that chroma is a viable feature for genre classification but also show that chroma using MFCC representation yields on average 10.68% better classification than using mean-based chroma.

	G_1	G_2	G_3	G_4	G_5	G_6	G_7	G_8	G_9	G_{10}
G_1	27	1	6	14	7	8	18	5	6	8
G_2	1	82	3	0	0	12	0	0	0	2
G_3	9	4	30	7	9	15	8	6	4	8
G_4	13	1	10	38	3	2	13	4	11	5
G_5	8	0	4	2	37	0	3	29	12	5
G_6	7	12	12	2	1	54	0	2	3	7
G_7	19	0	5	18	6	3	36	3	3	7
G_8	7	1	4	6	32	3	5	35	2	5
G_9	12	3	7	24	15	3	5	8	14	9
G_{10}	12	4	10	12	8	8	14	4	9	19

(a) MFCC representation

Figure 30: Confusion matrix using linear logistic regression models on MFCC and chroma-MFCC features.

9.1.2 *Discussion*

Figure 30 shows the confusion matrices for the classification of GTZAN genres using MFCC-based chroma. The row and column labels represent genre labels where: G_1

= Blues, G_2 = Classical, G_3 = Country, G_4 = Disco, G_5 = Hiphop, G_6 = Jazz, G_7 = Metal, G_8 = Pop, G_9 = Reggae, and G_{10} = Rock. Figure 30 is the result of classifying 10 GTZAN genres with MFCC-based chroma using random forests.

In Figure 30, there are many correctly classified classical and jazz pieces, which is consistent with our hypothesis. As expected classical music has been classified very well since classical music uses voice leading progressions which generally have a well defined soprano and bass voice. The four-part harmony is then completed by adding the third and fourth inner voices, and is usually terminated with a cadence. These cliché progressions are usually not adopted in more modern music and so chroma can very keenly classify classical music in this dataset which generally consists of more modern genres. This is evident in Figure 28 and Figure 29, where classical music is represented with green circles and classical pieces are further away from the centre of the cloud. Modern genres, such as pop, rock and R&B, use the same melodic phrasing and chordal structure. It important to note that although Reggae music consists of different instrumentation and rhythm patterns this is ignored by MFCC-based chroma, MFCC-chroma rather picks up the chordal and melodic progression similarities from Reggae as those by Rock, Pop and R&B genres.

	G_1	G_2	G_3	G_4	G_5	G_6	G_7	G_8	G_9	G_{10}
G_1	37	1	14	9	4	7	8	3	6	11
G_2	0	89	3	1	0	3	0	1	2	1
G_3	13	1	39	11	2	11	5	3	1	14
G_4	5	1	6	57	4	0	7	5	4	11
G_5	3	0	3	5	57	1	5	9	16	1
G_6	8	8	11	2	2	56	2	1	3	7
G_7	0	0	1	8	7	0	77	5	0	2
G_8	6	1	3	4	10	1	0	69	2	4
G_9	13	1	3	8	17	3	2	4	42	7
G_{10}	10	3	16	9	2	6	12	4	9	29

Figure 31: Confusion matrix using linear logistic regression models on MFCC and chroma-MFCC features for 10 GTZAN genres.

Example 9.2. As an example, Figure 29 shows a the classification of Jazz and disco GTZAN genres. An very common example of a Jazz progression is: *Cmaj7 Am7| Dm7 G7| Em7 A7| Dm7 and G7*. Some examples of music which make use of common progression is "Moose the Mooche" (by Pat Metheny), "Cheek to Cheek" (by George Van Eps), and "Shaw Nuff" (by Barney Kessel).

Since major and minor distonic chords contain many perfect fifths they are very commonly used in most modern genre composition. Diatonic melodies are also used in classical composition where harmony is an import aspect for many phrases. On the other hand, when using non-diatonic scales in Indian music composition, often there are no chordal changes present for MFCC-based chroma to detect. This lack of chordal changes using nondistonic scales have been observed in hard rock, hip hop [Pressing 2002], funk, disco, jazz, etc.

9.2 MFCC-BASED CHROMA VS. MEAN-BASED CHROMA

For both MFCC-based and mean-based chroma features, we used a standard mean-based 20 MFC coefficients, based on a 20-bin Mel-spectrum extending to 16 kHz.

Feature	Classifier	Accuracy	Time to build model
Test 1	Multilayer Perception	54.6%	13.51
	Naïve Bayes	47.5%	0.001
	Support Vector Machine	48.8%	1.03
	Linear Logistic Regression Models	55.2%	3.59
	Random Forest	48.7%	0.13

Table 24: Classification scores for both mean-based MFCC and MFCC-based chroma on 10 GTZAN genres.

Feature	Classifier	Accuracy	Time to build model
Test 2	Multilayer Perception	50.5%	5.33
	Naïve Bayes	34.9%	0.03
	Support Vector Machine	48.00%	0.98
	Linear Logistic Regression Models	51.6%	2.34
	Random Forest	49.9%	0.18

Table 25: Classification scores for both mean-based MFCC and mean-based chroma on 10 GTZAN genres.

The results modelling MFCC and mean -based chroma using our standard array of classification techniques we achieve the results presented by Table 24 and Table 25 respectively. From the results we see that all of the classification algorithms except random forests obtain slightly better classification scores for MFCC-based chroma. Combining MFCCs with MFCC-based chromes caused a statistically significant improvement and overall we obtain 3.98% better classification score by using MFCC-based compared to mean-based chroma. Although the multilayer perceptron produces a sufficient result in Table 24, it also takes a very long time to build. Linear logistic regression models produce a small but sufficient improvement with a quicker build time obtaining 55.2% classification of GTZAN genres. Figure 31 shows the confusion matrix of the linear logistic regression models' classification on MFCC-based chroma. It is clear that MFCC-based chroma and MFCC s quite effective when classifying Metal, Classical, Jazz, and Pop. It is also expected that these genres do not all share similar chordal progressions and lingual content.

9.3 CONCLUSION

The results confirm that MFCC-chroma provides 3.6% increase in precision with an overall 55.2% classification accuracy using only MFCC-based chroma than using mean-based chroma with MFCCs. Therefore, using MFCC-based chroma for genre classification not only increases precision but also serves as viable tool to effectively identify choral progressions. In order exploit the effectiveness of this feature, additional work is needed to transform chroma into a more effective tool to assess rotation and transpositions of chordal progressions that frequently appear in music.

Part III

MUSIC GENRE CLASSIFICATION

In this dissertation an exhaustive study for music genre classification is performed on 10 GTZAN genres. In this part the design matrix derived in the last four chapters is thinned to remove redundancy and dimensionality. In Chapter 10, only features that have a significant contribution to classify class labels are used to perform genre classification. Finally, in Chapter 11, the results of this dissertation are discussed and future work is presented.

10

AUTOMATIC MUSIC GENRE CLASSIFICATION

Music genre classification is the process where a piece of music is recognised, understood, and differentiated by a conventional category as belonging to a shared tradition or set of conventions [Cohen and Lefebvre 2005; Sadie 1980]. In this chapter, we use the information gain ranking algorithm for feature selection, to express graphically the contributions of each feature proposed in Chapter 4, and to reduce feature dimensionality. Using feature selection heuristics we select a cutoff point discarding redundant features and present the final design matrix with each feature's representation and dimensionality. Thereafter, the contributions of all previous chapters will be used to achieve a classification accuracy of 81% for 10 GTZAN genres using 10-fold cross validation with linear logistic regression models. Alongside this contribution we present the complete design matrix.

10.1 INTRODUCTION

Separating a non-linearly separable m-dimensional input vector linearly requires one to map the input vector to a higher dimension, as with higher dimensionality separability increases. However, in some cases, the dimensionality of the input vector is considered to be excessively large. In these cases, if the input vector is large and we want to map it non-linearly into a higher dimensional space then the size of the classifier will increases tremendously. Therefore, if the dimensionality of the input vector is unnecessarily large, we need to reduce its dimensionality for a better fit by the classification model. We can do this by detecting and removing redundancy present in the design matrix. One strategy is to start with many different features that are possibly relevant, and then out of that (given training samples), try and extract a good subset of features. This can be done by attempting to find the smallest subset of features that has the best correlation with the class labels using *information gain ranking*.

10.2 INFORMATION GAIN RANKING

Information gain ranking is a filter method that evaluates the worth of an attribute by measuring the information gain with respect to the class [Rai and Kumar 2014]. This is done by using the following formula:

$$\mathrm{InfoGain}(\mathrm{Class}, \mathrm{Attribute}) = \mathrm{H}(\mathrm{Class}) - \mathrm{H}(\mathrm{Class}|\mathrm{Attribute}). \tag{34}$$

Features which are labeled as insignificant or redundant in the design matrix can be considered as noise and only the remaining features will be used for classification. The information gain ranking algorithm organises features in terms of their

contribution so we can easily eliminate feature by simply selecting a cut-off point. To demonstrate this let $\vec{x}$ be an m-dimensional design matrix describing an element of an input vector where:

$$\vec{x} = (x_1, x_2, x_3, ..., x_{m_0}, x_{m_{0+1}}, ..., x_m). \tag{35}$$

Assuming the elements in $\vec{x}$ are ordered according to their information gain, if we wanted to reduce the dimensionality of $\vec{x}$ then we would simply use all of the features to the left of x_{m_0} (i.e. $\vec{x} = \{x_1, x_2, x_3, ..., x_{m_0}\}$) and discard all the features to the right of x_{m_0}. We assume that the sum of variances, or *mean square error*, of the elements eliminated to the right of x_{m_0} are insignificant or small. Figure 32 shows the results of using information gain ranking on the features in Table 4. The horizontal axis in Figure 32 corresponds to the feature numbers in Table 28, whereas the vertical axis corresponds to the contribution of that feature. Table 28 show the contributions of each feature in descending order. In order to eliminate features we need only choose a cutoff point in the scree plot given by Figure 32.

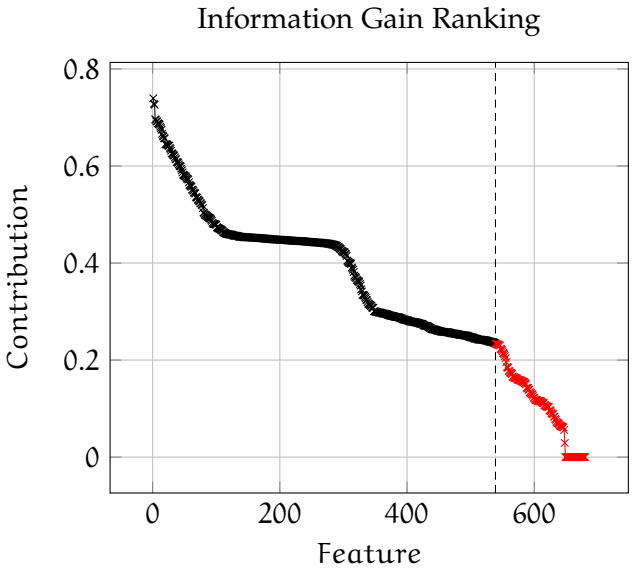


Figure 32: Information gain ranking of features based on contribution.

Choosing the first 539 features we present the following pruned design matrix in the first column of Table 26. The second column of Table 26 shows the eliminated features from the complete design matrix. The cut-off point was chosen by considering Figure 33, which shows the results of taking different numbers of features with the highest contributions and using them to classify 10 GTZAN genres. Although, we could have chosen up to 100 features and achieved between 70-75% classification accuracy, doing this would bias the learning model to this particular dataset since more diverse-genre databases might need these descriptors to achieve favourable classification.

Table 26: Information gain ranking: features contributions.

Features Maintained (459)	**Eliminated Feature (223)**
Spectral flux (MFCC) (4)	Peak flux (20-bin FH) (20)
Spectral variability (MFCC) (4)	Peak smoothness (mean) (1)
Compactness (mean + SD) (2)	Shape statistic centroid, skewness and kurtosis (mean) (3)
MFCCs (MFCC) (52)	Strongest frequency of centroid (MFCC) (4)
Peak centroid (mean + SD) (2)	Spectral rolloff (mean) (1)
Peak smoothness (SD) (1)	Strongest frequency of FFT (MFCC) (4)
Complex domain onset detection (mean) (1)	Envelope centroid, skewness and kurtosis (mean) (4)
Loudness (+ sharpness and spread) (mean) (26)	Beat histogram (mean) (171)
OBSI (+ radio) (mean) (17)	Strongest beat (mean + SD) (2)
Spectral decrease (mean) (1)	Strength of strongest beat (SD) (1)
Spectral flattness (mean) (20)	Fraction of low energy (SD) (1)
Spectral slope (mean) (1)	Beat sum (MFCC) (4)
Shape statistic spread (mean) (1)	Relative difference function (MFCC) (4)
Spectral centroid (MFCC) (4)	Temporal statistic centroid, skewness and kurtosis (mean) (3)
Spectral rolloff (SD) (1)	
Spectral crest (mean) (19)	
Spectral variation (mean) (1)	
Autocorrelation coefficients (mean) (49)	
Amplitude modulation (mean) (8)	
Zero crossing + SF (MFCC) (8)	
Envelope statistic spread (1)	
LPC and LSF (mean) (12)	
RMS (mean + SD) (2)	
Fraction of low energy (mean) (1)	
Beat histogram (SD) (171)	
Strength of strongest beat (mean) (1)	
Temporal statistic spread (mean) (1)	
Chroma (MFCC) (48)	

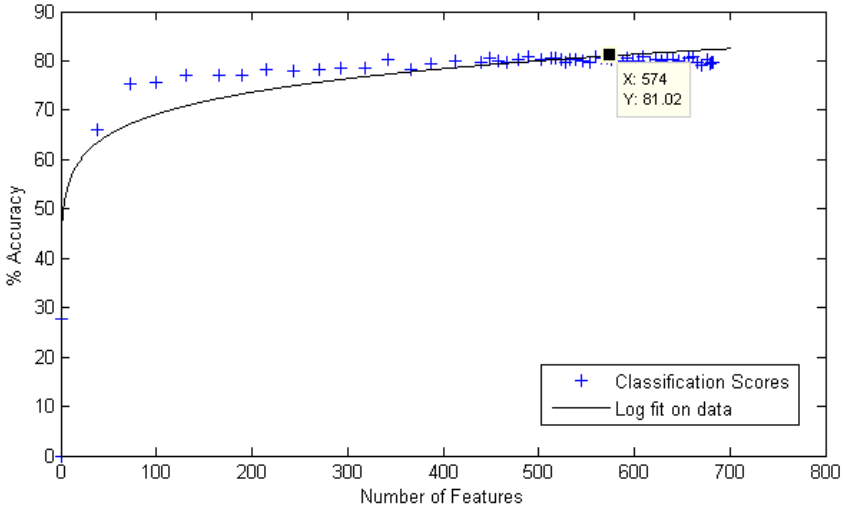


Figure 33: Relationship between the number of features and classification accuracy using IGR to guide feature trade-off.

In this section we effectively use all of the contributions from previous chapters to perform genre classification on 10 GTZAN genres. Table 30, in Appendix B of this dissertation, details the design matrix from Table 26 after information gain was performed. The second column in Table 30 provides the feature name; the third column provides the feature representation; the fourth column provides the feature dimension; and finally the fifth column details the parameters used when extracting the feature from the GTZAN genres. These feature parameters are provided to researchers who wish to replicate the experiments.

	G_1	G_2	G_3	G_4	G_5	G_6	G_7	G_8	G_9	G_{10}
G_1	84	0	3	3	0	5	1	0	2	2
G_2	0	96	1	0	0	2	0	0	0	1
G_3	3	0	77	2	0	4	0	1	3	10
G_4	1	1	5	76	2	0	0	4	5	6
G_5	1	0	0	1	85	0	4	3	6	0
G_6	3	4	5	1	0	82	1	2	1	1
G_7	2	0	0	1	1	0	90	0	0	6
G_8	0	0	4	4	1	0	0	84	1	6
G_9	2	0	3	6	6	1	1	4	70	7
G_{10}	5	0	7	9	2	0	5	5	1	66

Figure 34: 81% accuracy achieved with linear logistic regression models to classify 10-GTZAN genres using 10-fold cross validation.

Using the design matrix described in Table 30 we achieve the results in Table 27 which also shows the necessary time to build and evaluate each model. It is seen that although the multilayer perception takes a significant time to build and evaluate, it notably outperforms the naïve Bayes and support vector machine algorithms. The K-nearest neighbours and random forest algorithms take the least time to build and evaluate and produce sufficient results. The linear logistic regression model provides the best classification score. However, with the exception of the multilayer perceptron, the linear logistic regression model takes the longest time to build.

In real applications the time taken to build and evaluate the learning model needs to be considered carefully as a model will need to be retrained periodically should genre definitions continue to evolve. It is better to choose a classifier that separates genre spaces more effectively in the long run, than one that just provides excellent classification for a short time, however, this will entirely depend on the application. Table 29 provides the top 4 features that classify a particular genre well. These features have been selected based on their individual ability to classify each GTZAN genre with respect to the other 9 genres in Part ii.

Figure 34 shows the confusion matrix for 10 GTZAN genres using linear logistic regression models with 10-fold cross validation. The row and column labels represent genre labels where: G_1 = Blues, G_2 = Classical, G_3 = Country, G_4 = Disco, G_5 = Hiphop, G_6 = Jazz, G_7 = Metal, G_8 = Pop, G_9 = Reggae, and G_{10} = Rock. Again the fuzziness between rock and country music (and rock and disco) are observed.

Table 27: Automatic music genre classification of the thinned features vector.

Classifier	**Accuracy**	**Time to build model**	**Time to evaluate Model**
Naïve Bayes	46.40%	0.11 sec	2.13 sec
Support vector machines	32.50%	6.04 sec	38.12 sec
Multilayer perceptron	63.70%	635.37 sec	6 hours 20.12 sec
Linear logistic regression models	81.00%	20.25 sec	10 mins 31 secs
K-nearest neighbours	72.80%	0.02 sec	13.12 sec
Random forests	66.60%	0.22 sec	3.76 sec

10.4 CONCLUSION

Although recent classification precisions suggest that the performance of learning models for genre classification have become bounded, there is no confirmation to date that suggest these bounds cannot be exceeded. Nonetheless, small changes to existing models are unlikely to produce significantly better classification scores and so more attention to how feature extraction and classification are performed, or perhaps completely new approaches, are crucial to greatly excel through these bounds.

Using feature selection techniques that remove features based on variance, such as principal component analysis and information gain ranking, can disadvantage the application of the learning model conceptually. For example, the fraction of low energy feature, which had been removed in Section 10.2, could be very useful had the dataset

considered electronic dance and impressionistic music genres. Although, these feature selection algorithms can optimise classification precision for a specific dataset, one should consider that these features being eliminated were meant to describe specific aspects of music that might be explored better in other data. Therefore, imposing feature selection algorithms that maintain feature definitions instead of eliminating them should rather be considered[1]. Alternatively, running feature selection anew for each new dataset should be a preliminary requirement.

Given the variety of genre in current online music databases (MusicBrainz, Amazon, and BeeMp3) it is unlikely that they can be accurately represented by small datasets with 1000 songs and 10-genre labels[2]. Erroneous genre labels are often caused by inexperienced respondents; not being exposed to enough of the recording [Gjerdingen and Perrott 2008; Perrot and Gjerdigen 1999; Lippens *et al.* 2004]; and attempting to test the functionality of a learning model in a specific way rather than a more general way. The reliability of a learning model is purely measured by the quality of its ground truth and so extensive measures must be taken to ensure that the ground truth is well founded and motivated.

1 Such methods include genetic algorithm-based and forward-backward feature selection algorithms.
2 Such as GTZAN which contains 100 songs per genre.

11

CONCLUSION AND FUTURE WORK

Efforts in automatic music genre classification have only been producing minor improved success rates in recent literature. Being an arduous research direction, many authors believe that efforts should rather focus on devising a new method to enhance music organisation and browsing. In spite of these views, automatic music genre classification holds great significance both towards industry as well as to individual consumers. In this chapter we review the importance of genre classification and provide useful ideas for the enhancement of the research direction. Optimistically, this would inspire researchers to appreciate the potential of this problem and further encourage research in automatic music genre classification.

In this dissertation we analysed content-based features for music genre classification. As expressed in Section 1.2, since genre classification is usually performed by humans who observe cultural features (observations of arts and other manifestations of genre cognitively regarded collectively) more than content related features, we should not expect to achieve ground breaking results by classifying genre purely on content-based features. This is evident as the best genre classification algorithms using content-based features only achieve between 75-83% on 10 GTZAN genres. General ideas associated with fundamental musicology and genre, rather than concise regulations concerning genre, are essential concepts when composing or performing music. It can be further seen that the interest that many composers and performers have to convey culture far surpass their need to abide standard genre 'definitions' in music content itself. Thus, incorporating cultural features with structural ones in the feature domain could notably increase current classification rates [McKay and Fujinaga 2004; McKay 2004]. Although obtaining these cultural features can be inconvenient, Whitman and Smaragdis [2002] investigated the rewards of using web content mining for cultural features together with content-based features for music genre classification.

Large scale musical structures are present in most music genre types. Sometimes these structures are suppressed into the music by the composer, for example cyclic form and baroque dances, and other times the songwriter is unknowingly conditioned to a particular large scale structure, for example rondo (ABCBA) or even binary form (AABB). Understanding the form of a piece of music can immediately designate a small set of potential genre categories to which the piece could belong to. Most researchers who perform feature extraction for genre classification only calculate features based on local window intensities and disregard how a piece of music structurally changes over time. These overall structure-based feature descriptions can be preserved in learning models by using classifiers that exhibit memory[1]. Preserv-

1 Much like how hidden Markov models or recurrent neural networks work.

ing memory in learning models have been mostly ignored and could hold the key to better understanding chordal progressions and complex melodic structures.

Since feature extraction, selection, representation, and classification are all technically demanding tasks, researchers spend more time on these components and do not spend the time needed to compile quality datasets with masterful labelling. Ignoring this component compromise the performance of the entire learning model. Furthermore, many authors conveniently construct corpus datasets with genre labels that could be biased towards their study or completely inaccurate [Basili *et al.* 2004; Cilibrasi *et al.* 2004]. However, publicly available music genre datasets, such as GTZAN and Magnatune [MAG], themselves have a lot to be desired. Researchers should not just use publicly available music databases without significant attempts to rectify ambiguous labelling in the dataset. Furthermore, since the significance of the dataset is to provide a representative example of an actual popular music database, these datasets should contain recent and more popular music that the public are aware of.

Instead of trying to contain the entirety of genre diversity from these enormous databases into these relatively small datasets, one can rather construct datasets based on different characteristics that a genre learning model should exhibit and detect. For example, being able to detect instrumentation within classical music can be tested using a separate dataset altogether, whereas another dataset might test for structural form. It can be seen that the more datasets that are used to detect different characteristics in the learning model, the more dependable the learning model becomes. Someways to ensure the reliability of datasets include exposing these datasets to music genre experts; using surveys to assess genre labels; and web content mining to extract meta-data of recordings from the Internet. These three methods can be compared and a final decision will be made on the music genre label.

The musicality of a listener is not only required when constructing ground truth, but can also be used to satisfy a particular customer's genre preference. Further empirical research in human responses to genre classification can reveal if certain consumers with different musicality will appreciate music differently. Empirical research should compare and contrast different classification scores for different kinds of customers in terms of age, culture, and musicality. This type of psychological research will enhance our understanding of the possibilities to increase the dependability of ground truth and will also allow us to personalise multiple learning models to cater for groups of individuals' needs rather than forcing a one fits all approach.

Another key aspect to consider when labelling recordings is the list of genre labels used. Again, using music genre experts; surveys; and web content mining to obtain genre candidates for a set of recording could produce promising results as using 10 broad genres to classify a large set of recordings is impractical. Time should be taken to organise promising candidates corresponding to a recording that can be easily presented for empirical research. These candidates will include both broad genres as well as possible sub-genres. Furthermore, genre labels need to be thoroughly understood by the respondent, as once the ambiguity between genre definition have been somewhat cleared the respondent can participate to the best of their ability.

An additional problem with current genre labelling is placing albums and artists into genre catalogues. This causes every recording belonging to that said artist or album to inherit their genre label, which should not be the case. *Taylor Swift*, for

example, had been publicly labelled as a country singer, however, after publishing her *1989* album, in 2014, she made a 'genre jump' and declared that she will now be writing more pop music than county. Other examples of artists who have made 'genre jumps' include *Kate Nash* (who moved from pop to punk music); *Snoop Dogg* (who moved from rap to reggae and back to rap again); *Bob Dylan* (who moved from world to rock); and *Nellie Furtado* (who moved from Pop to Hiphop). Therefore, recordings which inherit genre labels from artists or albums are unrealistic and efforts need to be made to ensure that all recording are labelled independently.

A promising approach to music genre classification is performing multi-label automatic classification, which offers a solution to the fuzziness between genre definitions. Consequently, additional efforts need to be made to assign multiple labels to the ground truth. This could increase the burden on the respondent rendering fewer labelled tracks and genre subjectivity. Conversely, this could in fact accommodate respondents better as it would allow them to tag multiple genre labels on a single recording, should the piece of music resemble features from many different genres types, rather than being forced to chose just one. It would be even more useful to allow the respondent to tag an *assignment percentage* to each recording - should it belong to multiple genre labels. For example, *'Dear Mr. President'* by P!nk could be 50% soft rock, 25% acoustic, and 25% country rock. The intended purpose of these percentages tell us which genre is more noticeable in the music. For example, these *assignment percentages*, tagged on the song by P!nk above, gives us a good idea that the piece exhibits more soft rock properties than the other genre tags. This type of classification will enhance the calibre of ground truth, which is a major contribution in and of itself[2].

Understanding the strength of connections between genre tags could aid the construction of a weighted graph that maps genre to genre relationships. This type of ontological composition can not only help us understand how genres influence each other, but would also allow us to model structured classification strategies [McKay 2004]. This type of framework can account for differences in respondents' opinions when classifying music by genre, this is evident as studies by McKay [2004] show that organising structured classification strategies for genre, based on similarity rankings, can contribute considerably to better human classification rates. This model of genre relationships will help enhance music recommendation systems as customers will be able to search through large music databases based on the strength that their preferred genre has with respect to other genre types. For example, if a customer prefers progressive and disco music genres than the customer should also appreciate electronic dance genre types. Implementing this type of relationship will also allow customers who know more about specific genre labels to find other genres that might peek their interests by browsing through genre with high correlation corresponding to their preferred genre. For example, a customer who might be mildly interested in classical music might want to only explore classical pieces at a very high level, whereas a customer with a thorough background might rather prefer to browse a database for Baroque music that only have a certain type of counter-point or articulation.

2 This allows for more realistic implications.

Another noteworthy comment about automatic music genre classification concerns the type of mis-classifications that a learning model can achieve. For example, classifying some disco music as pop is less serious than classifying it as classical. As mentioned before, the definitions of genre are fuzzy, and to further complicate matters, the ground truth for genre classification models are not always of great quality. The genre interrelationship model, described above, can be used to account for these mis-classification and emphasise the closeness that some genres have between others. Furthermore, this genre relationship model can be integrated into the learning model to help guide classification decisions in some way both in the training as well as the final classification - perhaps some aspects of reinforcement learning could be useful. However, this type of model could be insensitive to music that displays different genre features in completely separate sections of the recording - which could be a good indicator for very high level genre labels.

Part IV

APPENDIX

A

FUNDAMENTAL MATHEMATICAL CONCEPTS

A.1 ROOT MEAN SQUARE

Definition A.1. *Root Mean Square*
The root mean square measures the magnitude of a varying quantity, which makes it very suitable for discrete-time signals. Let $\{y_1, y_2, y_3, y_4, ..., y_n\}$ be a set of arbitrary values, then the root mean square is calculated as:

$$y_{rms} = \sqrt{\frac{1}{n}(y_1^2 + y_2^2 + y_3^2 + ... + y_n^2)}. \tag{36}$$

Extending this definition for a DTS, $x(n)$, defined over the interval $T_1 \leqslant T_2$, leads to:

$$x_{rms} = \sqrt{\frac{1}{T_2 - T_1}\int_{T_1}^{T_2}[x(n)]^2 dt}. \tag{37}$$

Cartwright [2007] demonstrates how to calculate the RMS of a DTS without explicitly solving the integral.

A.2 ARITHMETIC MEAN

Definition A.2. *Arithmetic Mean*
The arithmetic mean is defined as the sum of a set of numbers divided by the set size [Jacobs 1994]. Let $S = \{x_1, x_2, x_3, ..., x_n\}$ containing n terms. Then the arithmetic mean of S is given as:

$$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i. \tag{38}$$

The arithmetic mean is commonly denoted as $\bar{x}_A$, where $\bar{x}_A$ is the arithmetic mean of the set $\{x_1, x_2, x_3, ..., x_n\}$ containing n terms [Medhi 1992].

A.3 GEOMETRIC MEAN

Definition A.3. *Geometric Mean*
The geometric mean is defined as the n^{th} root of a product of n numbers. Let $S = \{x_1, x_2, x_3, x_4, ..., x_n\}$ containing n terms. Then the geometric mean of S is given by:

$$\bar{x}_G = (\prod_{i=1}^{n} x_i)^{\frac{1}{n}} = \sqrt[n]{x_1 x_2 x_3 ... x_n}. \tag{39}$$

A.4 EUCLIDEAN DISTANCE

Definition A.4. *Euclidean Distance*
The Euclidean distance is defined as the linear segment that connects two points. If $p = (x_1, x_2, ..., x_n)$ and $q = (y_1, y_2, ..., y_n)$ are two points in n-dimensional space, then the Euclidean distance from x to y (or symmetrically y to x) is defined as:

$$D(x, y) = D(y, x) \tag{40}$$
$$= \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2 + ... + (x_n - y_n)^2} \tag{41}$$
$$= \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2} \tag{42}$$

Taking x as a positional vector in n-dimensions yields the following norm for x:

$$||x|| = \sqrt{x_1^2 + x_2^2 + ... + x_n^2}. \tag{43}$$

A.5 WEIGHTED MEAN

Definition A.5. *Weighted Mean*
Let $S = \{x_1, x_2, ..., x_n\}$ be a set where $S \neq \emptyset$ with non-negative weights $W = \{w_1, w_2, ..., w_n\}$. Then the weighted mean is defined as:

$$\bar{x}_W = \frac{\sum_{i=1}^{n} w_i x_i}{\sum_{i=1}^{n} w_i}, \tag{44}$$

where that the arithmetic mean is a special case, with $w_i = 1 \forall i$.

A.6 CONVOLUTION

Definition A.6. *Convolution*
The convolution of two functions, f and g, is defined as an integral transform by Equation 30, choosing $K = t - \tau$, convolution is defined as follows:

$$(f * g)(t) \tag{45}$$
$$= \int_{-\infty}^{\infty} f(t - \tau) g(\tau) d\tau \tag{46}$$
$$= \int_{-\infty}^{\infty} f(\tau) g(t - \tau) d\tau \tag{47}$$

The convolution is commonly written as $f * g$.

A.7 COMPLEX CONJUGATE

Definition A.7. *Complex Conjugate*
Complex conjugates is a pair of complex numbers, both having the same real part, but with imaginary parts of equal magnitude and opposite signs [Mathews and Walker 1970]. The conjugate of the complex number z: $z = x + iy$ and $\bar{z} = x - iy$, where $x, y \in \mathbb{R}$, are equivalent.

A.8 HANNING WINDOW

Definition A.8. *Hanning Window*
The Hanning function is a discrete window function typically used to choose a subset series of samples to perform a discrete-time Fourier transform operation. The Hanning window function is given by:

$$w(n) = 0.5(1 - \cos(\frac{2\pi n}{N-1})), \tag{48}$$

or

$$w(n) = \sin^2(\frac{\pi n}{N-1}). \tag{49}$$

B

ADDITIONAL TABLES AND FIGURES

Table 28: Information gain ranking: features contributions.

Contribution	Number	Feature
0.7395	1	Spectral flatness Coefficient 11
0.7283	2	MFCC: Chroma coefficient 36
0.7261	3	Compactness Average 0
0.696	4	MFCC: Chroma coefficient 12
0.6955	5	MFCC: Chroma coefficient 8
0.6921	6	MFCC: Chroma coefficient 0
0.6918	7	MFCC: Chroma coefficient 4
0.688	8	MFCC: Chroma coefficient 40
0.6875	9	MFCC: Chroma coefficient 28
0.6865	10	MFCC: Chroma coefficient 16
0.6849	11	Spectral flatness Coefficient 10
0.681	12	MFCC: Chroma coefficient 20
0.6781	13	MFCC: Chroma coefficient 24
0.6746	14	MFCC: Chroma coefficient 32
0.6663	15	MFCC: Chroma coefficient 44
0.6625	16	Spectral flatness Coefficient 9
0.6606	17	AutoCorrelation ACNbCoeff 24
0.6572	18	Spectral flatness Coefficient 13
0.6567	19	MFCC: MFCC8
0.6449	20	AutoCorrelation ACNbCoeff 28
0.6447	21	AutoCorrelation ACNbCoeff 30
0.6446	22	AutoCorrelation ACNbCoeff 29
0.6439	23	AutoCorrelation ACNbCoeff 26
0.6423	24	SpectralCrestFactorPerBand Coeff 9
0.6421	25	AutoCorrelation ACNbCoeff 25
0.6417	26	LPC Coeff 1
0.6374	27	Spectral flatness Coefficient 8
0.635	28	Spectral flatness Coefficient 12
0.6315	29	AutoCorrelation ACNbCoeff 21
0.6262	30	SpectralCrestFactorPerBand Coeff 11

Continued on next page

Table 28 – *Continued from previous page*

Contribution	Number	Feature
0.6255	31	SpectralCrestFactorPerBand Coeff 10
0.6235	32	Spectral flatness Coefficient 14
0.6218	33	MFCC: Spectral Fluxo
0.6215	34	AutoCorrelation ACNbCoeff 20
0.6176	35	Complex Domain Onset Detection
0.6142	36	SpectralCrestFactorPerBand Coeff 14
0.614	37	SpectralCrestFactorPerBand Coeff 13
0.6084	38	Amplitude Modulation: 4
0.6076	39	AutoCorrelation ACNbCoeff 31
0.607	40	SpectralCrestFactorPerBand Coeff 12
0.6035	41	AutoCorrelation ACNbCoeff 16
0.601	42	LPC Coeff 2
0.5996	43	AutoCorrelation ACNbCoeff 23
0.5987	44	MFCC: MFCC16
0.5946	45	AutoCorrelation ACNbCoeff 27
0.5912	46	SpectralFlatness
0.5868	47	MFCC: Spectral Variabilityo
0.5821	48	Amplitude Modulation: 2
0.5809	49	Spectral flatness Coefficient 7
0.5804	50	AutoCorrelation ACNbCoeff 18
0.5787	51	AutoCorrelation ACNbCoeff 22
0.5784	52	AutoCorrelation ACNbCoeff 32
0.5762	53	Spectral Shape Statistics: spread
0.5741	54	Amplitude Modulation: 7
0.5737	55	AutoCorrelation ACNbCoeff 13
0.572	56	OBSI Coeff 9
0.5664	57	Loudness Coeff 24
0.5611	58	AutoCorrelation ACNbCoeff 8
0.5597	59	AutoCorrelation ACNbCoeff 19
0.5584	60	AutoCorrelation ACNbCoeff 9
0.557	61	AutoCorrelation ACNbCoeff 10
0.5514	62	AutoCorrelation ACNbCoeff 11
0.5512	63	Root Mean Square Standard Deviationo
0.5489	64	MFCC: MFCC12
0.5453	65	SpectralCrestFactorPerBand Coeff 8
0.5445	66	Amplitude Modulation: 3
0.5432	67	AutoCorrelation ACNbCoeff 33

Continued on next page

Table 28 – *Continued from previous page*

Contribution	Number	Feature
0.5361	68	AutoCorrelation ACNbCoeff 14
0.535	69	MFCC: Strongest Frequency Via Zero Crossingso
0.5348	70	SpectralCrestFactorPerBand Coeff 7
0.5308	71	LSF 4
0.5301	72	AutoCorrelation ACNbCoeff 34
0.5279	73	AutoCorrelation ACNbCoeff 12
0.5269	74	AutoCorrelation ACNbCoeff 6
0.5232	75	AutoCorrelation ACNbCoeff 7
0.5232	76	AutoCorrelation ACNbCoeff 17
0.5202	77	AutoCorrelation ACNbCoeff 15
0.5193	78	Spectral flatness Coefficient 6
0.5132	79	AutoCorrelation ACNbCoeff 35
0.5042	80	SpectralCrestFactorPerBand Coeff 15
0.5026	81	Amplitude Modulation: 8
0.5001	82	AutoCorrelation ACNbCoeff 36
0.4986	83	Amplitude Modulation: 6
0.4984	84	Loudness Coeff 23
0.4973	85	AutoCorrelation ACNbCoeff 5
0.4961	86	LSF 10
0.4947	87	AutoCorrelation ACNbCoeff 4
0.4942	88	SpectralVariation
0.4929	89	AutoCorrelation ACNbCoeff 3
0.4922	90	AutoCorrelation ACNbCoeff 1
0.4916	91	Spectral Shape Statistics: centroid
0.4893	92	MFCC: Zero Crossingso
0.4869	93	LSF 9
0.4829	94	Spectral flatness Coefficient 15
0.4814	95	PerceptualSharpness
0.4805	96	Spectral flatness Coefficient 5
0.4801	97	MFCC: MFCC45
0.48	98	SpectralCrestFactorPerBand Coeff 6
0.4798	99	MFCC: MFCC4
0.4787	100	MFCC: MFCC41
0.4737	101	MFCC: MFCC1
0.4729	102	MFCC: MFCC0
0.472	103	AutoCorrelation ACNbCoeff 2
0.4712	104	AutoCorrelation ACNbCoeff 42

Continued on next page

Table 28 – *Continued from previous page*

Contribution	Number	Feature
0.4709	105	TemporalShapeStatistics spread
0.4699	106	Spectral flatness Coefficient 4
0.4694	107	AutoCorrelation ACNbCoeff 43
0.4681	108	Beat Histogram Average 148
0.4672	109	Beat Histogram Average 149
0.4664	110	Beat Histogram Average 147
0.4635	111	MFCC: MFCC37
0.4631	112	Beat Histogram Average 133
0.4622	113	Spectral Shape Statistics: skewness
0.4618	114	Beat Histogram Average 111
0.4608	115	Beat Histogram Average 164
0.4607	116	Beat Histogram Average 163
0.4598	117	Root Mean Square Average 0
0.4598	118	Beat Histogram Average 120
0.4594	119	MFCC: Spectral Centroid0
0.459	120	Beat Histogram Average 32
0.459	121	Beat Histogram Average 101
0.4586	122	Beat Histogram Average 116
0.4585	123	Beat Histogram Average 165
0.4584	124	Beat Histogram Average 67
0.4583	125	Beat Histogram Average 62
0.4574	126	LSF 3
0.4571	127	Beat Histogram Average 103
0.4564	128	Beat Histogram Average 167
0.4563	129	Beat Histogram Average 150
0.4562	130	Beat Histogram Average 102
0.4562	131	Beat Histogram Average 113
0.4561	132	Beat Histogram Average 140
0.4559	133	Beat Histogram Average 168
0.4557	134	Beat Histogram Average 146
0.4549	135	Beat Histogram Average 48
0.4544	136	Beat Histogram Average 31
0.4543	137	Beat Histogram Average 122
0.4541	138	Beat Histogram Average 114
0.454	139	Beat Histogram Average 99
0.454	140	Beat Histogram Average 119
0.4539	141	Beat Histogram Average 52

Continued on next page

Table 28 – *Continued from previous page*

Contribution	Number	Feature
0.4538	142	Beat Histogram Average 100
0.4532	143	Beat Histogram Average 145
0.4532	144	Beat Histogram Average 158
0.4531	145	Beat Histogram Average 33
0.453	146	Beat Histogram Average 93
0.453	147	Beat Histogram Average 135
0.453	148	Beat Histogram Average 157
0.4529	149	MFCC: Strongest Frequency Via Spectral Centroido
0.4529	150	Beat Histogram Average 97
0.4529	151	Beat Histogram Average 121
0.4527	152	Beat Histogram Average 49
0.4527	153	Beat Histogram Average 104
0.4527	154	Beat Histogram Average 156
0.4525	155	Beat Histogram Average 63
0.4524	156	Beat Histogram Average 61
0.4523	157	Beat Histogram Average 123
0.4522	158	Beat Histogram Average 64
0.4522	159	Beat Histogram Average 65
0.4521	160	Beat Histogram Average 26
0.4521	161	Beat Histogram Average 98
0.452	162	Loudness Coeff 22
0.4519	163	Beat Histogram Average 134
0.4516	164	Beat Histogram Average 46
0.4515	165	Beat Histogram Average 45
0.4515	166	Beat Histogram Average 144
0.4514	167	Beat Histogram Average 51
0.4513	168	Beat Histogram Average 42
0.4512	169	Beat Histogram Average 66
0.4511	170	Beat Histogram Average 115
0.4511	171	Beat Histogram Average 127
0.4509	172	Beat Histogram Average 53
0.4509	173	Beat Histogram Average 57
0.4508	174	Beat Histogram Average 112
0.4507	175	Beat Histogram Average 47
0.4504	176	Beat Histogram Average 136
0.4503	177	Beat Histogram Average 96
0.4501	178	Beat Histogram Average 128

Continued on next page

Table 28 – *Continued from previous page*

Contribution	Number	Feature
0.4501	179	Beat Histogram Average 131
0.4501	180	Beat Histogram Average 132
0.45	181	Beat Histogram Average 129
0.45	182	Beat Histogram Average 139
0.4494	183	Beat Histogram Average 34
0.4494	184	Beat Histogram Average 110
0.4492	185	Beat Histogram Average 40
0.4492	186	Beat Histogram Average 50
0.4492	187	Beat Histogram Average 56
0.449	188	OBSI Coeff 8
0.4488	189	Beat Histogram Average 83
0.4488	190	Beat Histogram Average 152
0.4487	191	Beat Histogram Average 143
0.4485	192	Beat Histogram Average 92
0.4484	193	Beat Histogram Average 130
0.4484	194	Beat Histogram Average 159
0.4482	195	Beat Histogram Average 24
0.4482	196	Beat Histogram Average 41
0.4482	197	Beat Histogram Average 55
0.4481	198	Beat Histogram Average 74
0.448	199	Beat Histogram Average 91
0.448	200	Beat Histogram Average 117
0.4479	201	Loudness Coeff 20
0.4479	202	Beat Histogram Average 124
0.4478	203	Beat Histogram Average 44
0.4477	204	Beat Histogram Average 16
0.4476	205	Beat Histogram Average 154
0.4476	206	Beat Histogram Average 160
0.4474	207	Beat Histogram Average 43
0.4474	208	Beat Histogram Average 58
0.4474	209	Beat Histogram Average 71
0.4473	210	Beat Histogram Average 27
0.4469	211	Beat Histogram Average 20
0.4469	212	Beat Histogram Average 81
0.4468	213	Beat Histogram Average 15
0.4466	214	Beat Histogram Average 94
0.4465	215	Beat Histogram Average 37

Continued on next page

Table 28 – *Continued from previous page*

Contribution	Number	Feature
0.4465	216	Beat Histogram Average 90
0.4465	217	Beat Histogram Average 107
0.4464	218	Beat Histogram Average 30
0.4462	219	Beat Histogram Average 151
0.4459	220	Spectral Rolloff Point Standard Deviationo
0.4458	221	Beat Histogram Average 155
0.4457	222	Beat Histogram Average 13
0.4457	223	Beat Histogram Average 25
0.4457	224	Beat Histogram Average 82
0.4456	225	Beat Histogram Average 95
0.4455	226	Beat Histogram Average 35
0.4455	227	Beat Histogram Average 60
0.4455	228	Beat Histogram Average 138
0.4454	229	Beat Histogram Average 39
0.4452	230	Beat Histogram Average 7
0.4451	231	LSF 5
0.4451	232	Beat Histogram Average 84
0.4451	233	Beat Histogram Average 109
0.4451	234	Beat Histogram Average 141
0.445	235	Beat Histogram Average 36
0.4448	236	Beat Histogram Average 23
0.4448	237	Beat Histogram Average 153
0.4447	238	Beat Histogram Average 108
0.4446	239	Beat Histogram Average 22
0.4443	240	Beat Histogram Average 69
0.444	241	Beat Histogram Average 19
0.444	242	Beat Histogram Average 21
0.444	243	Beat Histogram Average 106
0.4438	244	Beat Histogram Average 54
0.4438	245	Beat Histogram Average 80
0.4437	246	Beat Histogram Average 4
0.4435	247	Beat Histogram Average 8
0.4435	248	Beat Histogram Average 142
0.4433	249	Beat Histogram Average 59
0.4431	250	Beat Histogram Average 17
0.4431	251	Beat Histogram Average 118
0.4431	252	Beat Histogram Average 137

Continued on next page

Table 28 – *Continued from previous page*

Contribution	Number	Feature
0.443	253	Beat Histogram Average 18
0.4425	254	Beat Histogram Average 38
0.4425	255	Beat Histogram Average 166
0.4424	256	Beat Histogram Average 0
0.4424	257	Beat Histogram Average 86
0.4423	258	Beat Histogram Average 28
0.4423	259	Beat Histogram Average 125
0.4422	260	Beat Histogram Average 70
0.442	261	Beat Histogram Average 6
0.4419	262	Beat Histogram Average 89
0.4419	263	Beat Histogram Average 105
0.4418	264	Beat Histogram Average 5
0.4417	265	Beat Histogram Average 73
0.4415	266	Beat Histogram Average 126
0.4413	267	Beat Histogram Average 14
0.4412	268	Beat Histogram Average 12
0.4409	269	Beat Histogram Average 1
0.4409	270	Beat Histogram Average 3
0.4407	271	Beat Histogram Average 169
0.4406	272	Beat Histogram Average 29
0.4406	273	Beat Histogram Average 76
0.4404	274	Beat Histogram Average 87
0.4402	275	Beat Histogram Average 162
0.4397	276	LSF 8
0.4397	277	Beat Histogram Average 2
0.4397	278	Beat Histogram Average 85
0.4396	279	Beat Histogram Average 72
0.4389	280	Beat Histogram Average 68
0.4384	281	Beat Histogram Average 9
0.4383	282	Partial Based Spectral Centroid Standard Deviationo
0.4383	283	Beat Histogram Average 10
0.4377	284	Beat Histogram Average 11
0.4376	285	Beat Histogram Average 75
0.4371	286	Beat Histogram Average 79
0.4366	287	Beat Histogram Average 170
0.4362	288	Beat Histogram Average 88
0.4354	289	OBSIR Step 7

Continued on next page

Table 28 – *Continued from previous page*

Contribution	Number	Feature
0.4346	290	Beat Histogram Average 161
0.4327	291	Beat Histogram Average 78
0.4325	292	MFCC: MFCC49
0.4323	293	Beat Histogram Average 77
0.4305	294	AutoCorrelation ACNbCoeff 47
0.4296	295	Spectral flatness Coefficient 3
0.4285	296	MFCC: Beat Sumo
0.4259	297	MFCC: Spectral Variability1
0.4242	298	AutoCorrelation ACNbCoeff 40
0.4237	299	AutoCorrelation ACNbCoeff 46
0.4228	300	OBSI Coeff 2
0.422	301	AutoCorrelation ACNbCoeff 44
0.4219	302	AutoCorrelation ACNbCoeff 41
0.4194	303	OBSI Coeff 3
0.4177	304	AutoCorrelation ACNbCoeff 45
0.4146	305	SpectralCrestFactorPerBand Coeff 5
0.405	306	AutoCorrelation ACNbCoeff 49
0.403	307	AutoCorrelation ACNbCoeff 48
0.4019	308	OBSI Coeff 1
0.4001	309	MFCC: MFCC33
0.3997	310	Loudness Coeff 6
0.3994	311	LSF 7
0.3982	312	LSF 6
0.391	313	TemporalShapeStatistics kurtosis
0.389	314	Spectral flatness Coefficient 2
0.3862	315	MFCC: MFCC20
0.3814	316	OBSI Coeff 4
0.374	317	MFCC: MFCC24
0.3715	318	LSF 2
0.3714	319	Loudness Coeff 21
0.369	320	AutoCorrelation ACNbCoeff 37
0.3654	321	Loudness Coeff 19
0.3647	322	AutoCorrelation ACNbCoeff 38
0.3628	323	OBSIR Step 8
0.3621	324	Spectral Shape Statistics: kurtosis
0.3606	325	AutoCorrelation ACNbCoeff 39
0.3569	326	MFCC: MFCC25

Continued on next page

Table 28 – *Continued from previous page*

Contribution	Number	Feature
0.3535	327	MFCC: MFCC29
0.3458	328	PerceptualSpread
0.3415	329	Loudness Coeff 5
0.3345	330	Beat Histogram Standard Deviation157
0.3341	331	OBSIR Step 6
0.3334	332	MFCC: MFCC2
0.3331	333	Loudness Coeff 18
0.3301	334	Peak Based Spectral Smoothness Standard Deviationo
0.3276	335	MFCC: Spectral Flux2
0.3241	336	Beat Histogram Standard Deviation139
0.3234	337	Beat Histogram Standard Deviation133
0.3198	338	Partial Based Spectral Centroid Average 0
0.3164	339	Spectral flatness Coefficient 19
0.3158	340	SpectralCrestFactorPerBand Coeff 2
0.314	341	SpectralCrestFactorPerBand Coeff 4
0.3133	342	Beat Histogram Standard Deviation154
0.312	343	Beat Histogram Standard Deviation132
0.3116	344	OBSI Coeff 7
0.3084	345	Beat Histogram Standard Deviation160
0.3078	346	MFCC: MFCC6
0.3075	347	SpectralDecrease
0.3007	348	Beat Histogram Standard Deviation134
0.3004	349	Beat Histogram Standard Deviation158
0.3002	350	Beat Histogram Standard Deviation123
0.2996	351	Beat Histogram Standard Deviation162
0.2993	352	Beat Histogram Standard Deviation143
0.2991	353	Beat Histogram Standard Deviation170
0.2986	354	Beat Histogram Standard Deviation126
0.2982	355	Beat Histogram Standard Deviation135
0.2979	356	MFCC: Zero Crossings2
0.2979	357	MFCC: Strongest Frequency Via Zero Crossings2
0.2979	358	Beat Histogram Standard Deviation121
0.2974	359	Beat Histogram Standard Deviation161
0.2973	360	Beat Histogram Standard Deviation156
0.2962	361	Loudness Coeff 4
0.296	362	Beat Histogram Standard Deviation122
0.296	363	Beat Histogram Standard Deviation169

Continued on next page

Table 28 – *Continued from previous page*

Contribution	Number	Feature
0.2955	364	Spectral flatness Coefficient 16
0.2955	365	Beat Histogram Standard Deviation165
0.2951	366	Beat Histogram Standard Deviation144
0.2947	367	MFCC: MFCC42
0.2947	368	Beat Histogram Standard Deviation131
0.2942	369	Beat Histogram Standard Deviation155
0.2933	370	Beat Histogram Standard Deviation166
0.293	371	Beat Histogram Standard Deviation138
0.2924	372	Beat Histogram Standard Deviation125
0.292	373	Beat Histogram Standard Deviation140
0.2915	374	Beat Histogram Standard Deviation163
0.2912	375	Beat Histogram Standard Deviation142
0.291	376	Beat Histogram Standard Deviation141
0.2909	377	Beat Histogram Standard Deviation124
0.2909	378	Beat Histogram Standard Deviation146
0.2901	379	Beat Histogram Standard Deviation149
0.2896	380	Beat Histogram Standard Deviation127
0.2894	381	Beat Histogram Standard Deviation145
0.2888	382	Spectral flatness Coefficient 1
0.2887	383	SpectralCrestFactorPerBand Coeff 19
0.2887	384	Beat Histogram Standard Deviation137
0.2885	385	Beat Histogram Standard Deviation153
0.2884	386	Beat Histogram Standard Deviation148
0.2882	387	Beat Histogram Standard Deviation119
0.2875	388	Beat Histogram Standard Deviation120
0.2865	389	Beat Histogram Standard Deviation147
0.2861	390	Beat Histogram Standard Deviation159
0.2859	391	Beat Histogram Standard Deviation150
0.2857	392	Beat Histogram Standard Deviation136
0.2843	393	MFCC: MFCC44
0.2839	394	Beat Histogram Standard Deviation111
0.2836	395	Beat Histogram Standard Deviation151
0.2833	396	Beat Histogram Standard Deviation71
0.2832	397	Beat Histogram Standard Deviation114
0.2825	398	Beat Histogram Standard Deviation112
0.2823	399	Beat Histogram Standard Deviation128
0.2818	400	Beat Histogram Standard Deviation115

Continued on next page

Table 28 – *Continued from previous page*

Contribution	Number	Feature
0.2818	401	Beat Histogram Standard Deviation130
0.2814	402	Beat Histogram Standard Deviation94
0.2802	403	OBSIR Step 3
0.2801	404	SpectralCrestFactorPerBand Coeff 3
0.2798	405	Beat Histogram Standard Deviation118
0.2796	406	Beat Histogram Standard Deviation168
0.2792	407	Spectral flatness Coefficient 17
0.2789	408	SpectralCrestFactorPerBand Coeff 16
0.2789	409	Beat Histogram Standard Deviation129
0.2787	410	Beat Histogram Standard Deviation109
0.2782	411	MFCC: MCC28
0.278	412	Beat Histogram Standard Deviation152
0.2779	413	Beat Histogram Standard Deviation167
0.2778	414	Beat Histogram Standard Deviation95
0.2774	415	SpectralCrestFactorPerBand Coeff 17
0.2771	416	Beat Histogram Standard Deviation164
0.2764	417	Beat Histogram Standard Deviation107
0.2764	418	Beat Histogram Standard Deviation110
0.2757	419	Beat Histogram Standard Deviation83
0.2755	420	Beat Histogram Standard Deviation85
0.2748	421	Beat Histogram Standard Deviation113
0.2747	422	Beat Histogram Standard Deviation93
0.2747	423	Beat Histogram Standard Deviation116
0.2746	424	OBSIR Step 1
0.2732	425	Beat Histogram Standard Deviation117
0.2723	426	Beat Histogram Standard Deviation82
0.2713	427	Beat Histogram Standard Deviation108
0.2708	428	Beat Histogram Standard Deviation86
0.2708	429	Beat Histogram Standard Deviation106
0.2707	430	Beat Histogram Standard Deviation102
0.2705	431	Beat Histogram Standard Deviation87
0.27	432	Beat Histogram Standard Deviation103
0.2692	433	Strength Of Strongest Beat Standard Deviationo
0.2679	434	Compactness Standard Deviationo
0.2673	435	MFCC: MFCC48
0.2668	436	Beat Histogram Standard Deviation104
0.2665	437	Beat Histogram Standard Deviation96

Continued on next page

Table 28 – *Continued from previous page*

Contribution	Number	Feature
0.2645	438	Beat Histogram Standard Deviation68
0.2637	439	Beat Histogram Standard Deviation73
0.2633	440	Beat Histogram Standard Deviation81
0.2633	441	Beat Histogram Standard Deviation98
0.2632	442	Loudness Coeff 9
0.2629	443	Beat Histogram Standard Deviation97
0.2625	444	Beat Histogram Standard Deviation67
0.2622	445	Beat Histogram Standard Deviation84
0.2615	446	OBSIR Step 4
0.2611	447	Beat Histogram Standard Deviation65
0.2609	448	Beat Histogram Standard Deviation92
0.2606	449	Beat Histogram Standard Deviation60
0.2604	450	Beat Histogram Standard Deviation66
0.26	451	Beat Histogram Standard Deviation105
0.2597	452	Beat Histogram Standard Deviation54
0.2589	453	Beat Histogram Standard Deviation90
0.2588	454	Beat Histogram Standard Deviation99
0.2584	455	Beat Histogram Standard Deviation75
0.2584	456	Beat Histogram Standard Deviation88
0.2581	457	Beat Histogram Standard Deviation77
0.258	458	Beat Histogram Standard Deviation51
0.258	459	Beat Histogram Standard Deviation91
0.2578	460	MFCC: MFCC32
0.2578	461	Beat Histogram Standard Deviation8
0.2575	462	Beat Histogram Standard Deviation63
0.2574	463	Beat Histogram Standard Deviation13
0.2573	464	Beat Histogram Standard Deviation74
0.2572	465	Beat Histogram Standard Deviation53
0.2565	466	MFCC: MFCC5
0.2565	467	Beat Histogram Standard Deviation72
0.256	468	SpectralCrestFactorPerBand Coeff 18
0.2554	469	Beat Histogram Standard Deviation58
0.2554	470	Beat Histogram Standard Deviation101
0.255	471	Beat Histogram Standard Deviation57
0.2548	472	Beat Histogram Standard Deviation0
0.2548	473	Beat Histogram Standard Deviation7
0.2548	474	Beat Histogram Standard Deviation52

Continued on next page

Table 28 – *Continued from previous page*

Contribution	Number	Feature
0.2543	475	Beat Histogram Standard Deviation100
0.2542	476	Beat Histogram Standard Deviation69
0.2538	477	MFCC: MFCC38
0.2538	478	Beat Histogram Standard Deviation64
0.2535	479	Beat Histogram Standard Deviation50
0.2533	480	Beat Histogram Standard Deviation49
0.2526	481	Beat Histogram Standard Deviation78
0.2525	482	Beat Histogram Standard Deviation41
0.2522	483	Beat Histogram Standard Deviation80
0.2517	484	Beat Histogram Standard Deviation59
0.2517	485	Beat Histogram Standard Deviation76
0.2516	486	Beat Histogram Standard Deviation22
0.2516	487	Beat Histogram Standard Deviation79
0.2514	488	Beat Histogram Standard Deviation55
0.2512	489	Beat Histogram Standard Deviation1
0.2509	490	Beat Histogram Standard Deviation5
0.2507	491	Beat Histogram Standard Deviation6
0.2507	492	Beat Histogram Standard Deviation56
0.2505	493	Beat Histogram Standard Deviation40
0.2501	494	Beat Histogram Standard Deviation9
0.25	495	Beat Histogram Standard Deviation23
0.2494	496	Beat Histogram Standard Deviation14
0.2487	497	Beat Histogram Standard Deviation25
0.2479	498	Beat Histogram Standard Deviation26
0.2478	499	MFCC: MFCC46
0.247	500	Loudness Coeff 17
0.2469	501	Beat Histogram Standard Deviation70
0.2467	502	Beat Histogram Standard Deviation61
0.2466	503	Beat Histogram Standard Deviation11
0.2461	504	Beat Histogram Standard Deviation17
0.2459	505	Beat Histogram Standard Deviation48
0.2445	506	Beat Histogram Standard Deviation16
0.2444	507	Beat Histogram Standard Deviation24
0.2443	508	Beat Histogram Standard Deviation4
0.244	509	Beat Histogram Standard Deviation18
0.2437	510	Beat Histogram Standard Deviation89
0.2435	511	Beat Histogram Standard Deviation43

Continued on next page

Table 28 – *Continued from previous page*

Contribution	Number	Feature
0.2422	512	Loudness Coeff 8
0.2422	513	Beat Histogram Standard Deviation10
0.2422	514	Beat Histogram Standard Deviation27
0.2422	515	Beat Histogram Standard Deviation42
0.2421	516	Beat Histogram Standard Deviation12
0.2417	517	MFCC: MFCC21
0.2416	518	OBSI Coeff 5
0.2415	519	Beat Histogram Standard Deviation3
0.2414	520	Beat Histogram Standard Deviation2
0.2412	521	Beat Histogram Standard Deviation39
0.2409	522	MFCC: MFCC10
0.2406	523	Beat Histogram Standard Deviation38
0.2405	524	Beat Histogram Standard Deviation15
0.2405	525	Beat Histogram Standard Deviation19
0.2393	526	Beat Histogram Standard Deviation21
0.2392	527	Beat Histogram Standard Deviation29
0.2392	528	Beat Histogram Standard Deviation62
0.2382	529	Beat Histogram Standard Deviation28
0.2382	530	Beat Histogram Standard Deviation30
0.2378	531	Loudness Coeff 16
0.2375	532	Beat Histogram Standard Deviation33
0.2374	533	Beat Histogram Standard Deviation44
0.2366	534	Beat Histogram Standard Deviation31
0.2366	535	Beat Histogram Standard Deviation34
0.2366	536	Beat Histogram Standard Deviation47
0.2355	537	Beat Histogram Standard Deviation36
0.2353	538	Envelope Shape Statistics: Spread
0.2347	539	Histogram: Partial Based Spectral Flux19
0.2336	540	Beat Histogram Standard Deviation32
0.2326	541	Beat Histogram Standard Deviation45
0.2322	542	Beat Histogram Standard Deviation37
0.2319	543	OBSI Coeff 6
0.2318	544	Beat Histogram Standard Deviation20
0.2305	545	Beat Histogram Standard Deviation35
0.23	546	Beat Histogram Standard Deviation46
0.2233	547	SpectralCrestFactorPerBand Coeff 1
0.2222	548	Peak Based Spectral Smoothness Average 0

Continued on next page

Table 28 – *Continued from previous page*

Contribution	Number	Feature
0.2211	549	Spectral Rolloff Point Average 0
0.2185	550	Loudness Coeff 1
0.2146	551	MFCC: MFCC50
0.2137	552	MFCC: MFCC34
0.2116	553	MFCC: Chroma coefficient 42
0.2083	554	MFCC: MFCC40
0.2039	555	MFCC: MFCC36
0.1956	556	MFCC: Relative Difference Function2
0.1855	557	Strongest Beat Average 0
0.1847	558	Loudness Coeff 10
0.1843	559	Loudness Coeff 3
0.1782	560	MFCC: Chroma coefficient 45
0.1772	561	MFCC: Chroma coefficient 38
0.1745	562	MFCC: Chroma coefficient 34
0.1736	563	MFCC: Chroma coefficient 26
0.1728	564	Amplitude Modulation: 1
0.1713	565	Loudness Coeff 15
0.1661	566	MFCC: Chroma coefficient 2
0.1651	567	MFCC: Chroma coefficient 18
0.1643	568	MFCC: MFCC13
0.1636	569	Loudness Coeff 2
0.1634	570	MFCC: Chroma coefficient 46
0.163	571	MFCC: MFCC17
0.163	572	MFCC: Chroma coefficient 10
0.1612	573	MFCC: MFCC14
0.1596	574	MFCC: Chroma coefficient 30
0.159	575	MFCC: Spectral Centroid2
0.159	576	MFCC: Strongest Frequency Via Spectral Centroid2
0.1589	577	MFCC: Chroma coefficient 33
0.1586	578	MFCC: Chroma coefficient 37
0.1571	579	MFCC: Chroma coefficient 25
0.157	580	MFCC: Chroma coefficient 17
0.156	581	Loudness Coeff 7
0.1554	582	Spectral flatness Coefficient 18
0.1552	583	MFCC: Chroma coefficient 14
0.1547	584	MFCC: Chroma coefficient 6
0.1527	585	MFCC: Chroma coefficient 22

Continued on next page

Table 28 – *Continued from previous page*

Contribution	Number	Feature
0.152	586	OBSIR Step 5
0.1499	587	MFCC: Spectral Flux1
0.143	588	MFCC: Spectral Centroid1
0.143	589	MFCC: Strongest Frequency Via Spectral Centroid1
0.1422	590	MFCC: MFCC18
0.1395	591	MFCC: MFCC9
0.1387	592	Strength Of Strongest Beat Average 0
0.1339	593	OBSIR Step 2
0.1319	594	LSF 1
0.1316	595	TemporalShapeStatistics skewness
0.1294	596	MFCC: Strongest Frequency Via FFT Maximum1
0.1293	597	Loudness Coeff 11
0.1267	598	MFCC: MFCC35
0.1225	599	MFCC: Relative Difference Function3
0.1221	600	MFCC: MFCC43
0.119	601	TemporalShapeStatistics centroid
0.1176	602	MFCC: MFCC51
0.1175	603	MFCC: Zero Crossings3
0.1175	604	MFCC: Strongest Frequency Via Zero Crossings3
0.1162	605	MFCC: MFCC26
0.1155	606	MFCC: Spectral Centroid3
0.1155	607	MFCC: Strongest Frequency Via Spectral Centroid3
0.1153	608	MFCC: MFCC22
0.1153	609	MFCC: MFCC30
0.1148	610	MFCC: MFCC7
0.1148	611	MFCC: Chroma coefficient 21
0.1132	612	MFCC: MFCC39
0.1118	613	MFCC: MFCC31
0.1115	614	MFCC: Strongest Frequency Via FFT Maximum0
0.1111	615	MFCC: Chroma coefficient 41
0.1072	616	MFCC: MFCC27
0.1069	617	MFCC: Chroma coefficient 29
0.1051	618	MFCC: MFCC47
0.1051	619	MFCC: Chroma coefficient 13
0.1039	620	MFCC: Chroma coefficient 1
0.1038	621	MFCC: Chroma coefficient 5
0.1035	622	Fraction Of Low Energy Windows Standard Deviation0

Continued on next page

Table 28 – *Continued from previous page*

Contribution	Number	Feature
0.103	623	MFCC: Chroma coefficient 9
0.0966	624	MFCC: Spectral Variability2
0.0959	625	MFCC: Beat Sum1
0.0942	626	Loudness Coeff 12
0.0909	627	Loudness Coeff 13
0.0893	628	MFCC: Zero Crossings1
0.0893	629	MFCC: Strongest Frequency Via Zero Crossings1
0.0887	630	MFCC: MFCC3
0.0836	631	MFCC: MFCC11
0.0833	632	MFCC: Relative Difference Function1
0.0784	633	Fraction Of Low Energy Windows Average 0
0.0768	634	MFCC: MFCC15
0.0728	635	MFCC: MFCC19
0.0712	636	MFCC: Spectral Variability3
0.0699	637	MFCC: MFCC23
0.0695	638	MFCC: Strongest Frequency Via FFT Maximum3
0.0677	639	Histogram: Partial Based Spectral Flux13
0.0658	640	MFCC: Strongest Frequency Via FFT Maximum2
0.0642	641	Histogram: Partial Based Spectral Flux15
0.0637	642	Histogram: Partial Based Spectral Flux12
0.0637	643	Loudness Coeff 14
0.0623	644	Histogram: Partial Based Spectral Flux14
0.0618	645	Strongest Beat Standard Deviation0
0.0611	646	MFCC: Chroma coefficient 15
0.0559	647	Histogram: Partial Based Spectral Flux17
0.0291	648	MFCC: Relative Difference Function0
0	649	MFCC: Spectral Flux3
0	650	Histogram: Partial Based Spectral Flux0
0	651	Histogram: Partial Based Spectral Flux1
0	652	Histogram: Partial Based Spectral Flux2
0	653	Histogram: Partial Based Spectral Flux3
0	654	Histogram: Partial Based Spectral Flux4
0	655	Histogram: Partial Based Spectral Flux5
0	656	Histogram: Partial Based Spectral Flux6
0	657	Histogram: Partial Based Spectral Flux7
0	658	Histogram: Partial Based Spectral Flux8
0	659	Histogram: Partial Based Spectral Flux9

Continued on next page

Table 28 – *Continued from previous page*

Contribution	Number	Feature
0	660	Histogram: Partial Based Spectral Flux10
0	661	Histogram: Partial Based Spectral Flux11
0	662	Histogram: Partial Based Spectral Flux16
0	663	Histogram: Partial Based Spectral Flux18
0	664	Amplitude Modulation: 5
0	665	Envelope Shape Statistics: Centroid
0	666	Envelope Shape Statistics: Skewness
0	667	Envelope Shape Statistics: Kurtosis
0	668	MFCC: Beat Sum2
0	669	MFCC: Beat Sum3
0	670	MFCC: Chroma coefficient 3
0	671	MFCC: Chroma coefficient 7
0	672	MFCC: Chroma coefficient 11
0	673	MFCC: Chroma coefficient 19
0	674	MFCC: Chroma coefficient 23
0	675	MFCC: Chroma coefficient 27
0	676	MFCC: Chroma coefficient 31
0	677	MFCC: Chroma coefficient 35
0	678	MFCC: Chroma coefficient 39
0	679	MFCC: Chroma coefficient 43
0	680	MFCC: Chroma coefficient 47

Table 30: Feature list with each feature's representation; number of dimensions; and parameters.

No.	Feature Name	Rep.	Dim.	Parameters
1	Spectral Flux	MFCCs	4	Window size = 512 (no overlap); 16 kHz size rate.
2	Spectral Variability	MFCC	4	Window size = 512 (no overlap); 16 kHz size rate.
3	Compactness	Mean + SD	2	Window size = 512 (no overlap); 16 kHz size rate.
4	MFCCs	MFCC	52	Window size = 512 (no overlap); 16 kHz sample rate; 13 coefficients.
5	Peak Centroid	Mean + SD	2	Window size = 512 (no overlap); 16 kHz size rate;

Continued on next page

Table 30 – *Continued from previous page*

No.	Feature Name	Rep.	Dim.	Parameters
6	Peak Smoothness	SD	1	Window size = 512 (no overlap); 16 kHz size rate;
7	Complex Domain Onset Detection	Mean	1	Window size = 512 (no overlap); 16 kHz size rate; using Hanning window for FFT transform; output frame size = 1024.
8	Loudness (+ Sharpness and Spread)	Mean	26	Window size = 512 (no overlap); 16 kHz size rate; using Hanning window for FFT transform; normalising each band to sum to 1; output frame size = 1024.
9	OBSI (+ Radio)	Mean	17	Window size = 512 (no overlap); 16 kHz size rate; using Hanning window for FFT transform; minimum frequency for OBSI filter = 27.5; output frame size = 1024.
10	Spectral Decrease	Mean	1	Window size = 512 (no overlap); 16 kHz size rate; using Hanning window for FFT transform; output frame size = 1024.
11	Spectral Flattness	Mean	20	Window size = 512 (no overlap); 16 kHz size rate; using Hanning window for FFT transform; output frame size = 1024.
12	Spectral Slope	Mean	1	Window size = 512 (no overlap); 16 kHz size rate; using Hanning window for FFT transform; output frame size = 1024.
13	Shape Statistic spread	Mean	1	Window size = 512 (no overlap); 16 kHz size rate; using Hanning window for FFT transform; output frame size = 1024.
14	Spectral Centroid	MFCCs	4	Window size = 512 (no overlap); 16 kHz size rate.

Continued on next page

Table 30 – *Continued from previous page*

No.	Feature Name	Rep.	Dim.	Parameters
15	Spectral Rolloff	SD	1	Window size = 512 (no overlap); 16 kHz size rate; 85.5% cutoff point.
16	Spectral Crest	Mean	19	Window size = 512 (no overlap); 16 kHz size rate; using Hanning window for FFT transform; output frame size = 1024.
17	Spectral Variation	Mean	1	Window size = 512 (no overlap); 16 kHz size rate; using Hanning window for FFT transform; output frame size = 1024.
18	Autocorrelation Coefficients	Mean	49	Window size = 512 (no overlap); 16 kHz size rate; number of coefficients = 49; output frame size = 1024.
19	Amplitude Modulation	Mean	8	Window size = 16384 (no overlap); 16 kHz size rate; Decimation factor to compute envelope = 200; output frame size = 32 768.
20	Zero Crossing + SF	MFCC	8	Window size = 512 (no overlap); 16 kHz size rate.
21	Envelope Statistic Spread	Mean	1	Window size = 16384 (no overlap); 16 kHz size rate; Decimation factor to compute envelope = 200; output frame size = 32 768.
22	LPC and LSF	Mean	12	Window size = 512 (no overlap); 16 kHz size rate; number of coefficients = 2; output frame size = 1024.
23	RMS	Mean + SD	2	Window size = 512 (no overlap); 16 kHz size rate.
24	Fraction of Low Energy	Mean	1	Window size = 512 (no overlap); 16 kHz size rate; n = 100 (see Section 7.2).
25	Beat Histogram	SD	171	Window size = 512 (no overlap); 16 kHz size rate.
26	Strength of Strongest Beat	Mean	1	Window size = 512 (no overlap); 16 kHz size rate.

Continued on next page

Table 30 – *Continued from previous page*

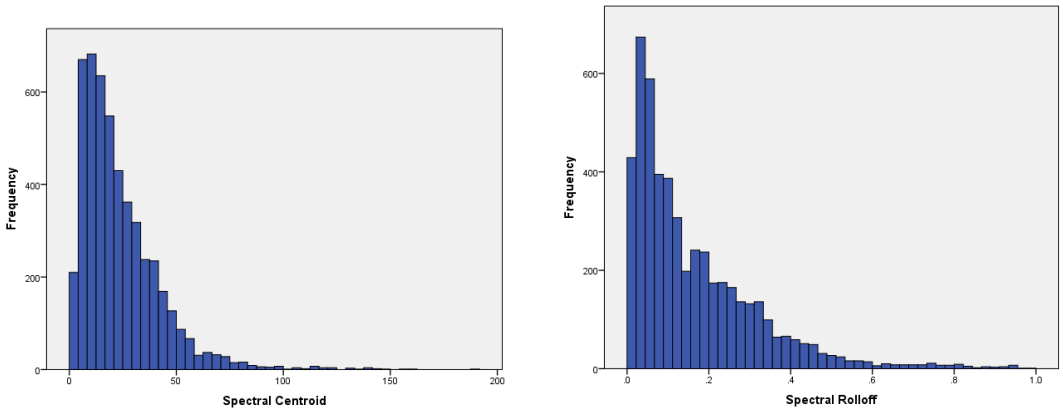
No.	Feature Name	Rep.	Dim.	Parameters
27	Temporal Statistic Spread	Mean	1	Window size = 512 (no overlap); 16 kHz size rate;
28	Chroma	MFCC	48	Window size = 512 (no overlap); 16 kHz size rate; output frame size = 1024.

Table 31: Parameters of classifiers used.

No.	Feature Name
SVM	SVMType C-SVC (classification); CacheSize 40.0; coefo 0.0; cost 1.0; debug false; degree 3; replace missing values; eps 0.001; gamma 0.0; kernalType radial basis function: $\exp(-gamma * \|u - v\|^2)$; loss 0.1; normalise false; nu 0.5; probability Estimates false; seed 1: shrinking true
LLRM	Heuristic Stop 50; Max boosting iterations 500; Num boosting iterations 0
MP	Hidden Layer a; learning rate 0.3; momentum 0.2; seed 0; training time 500; validation set size 0; validation threshold 20
KNN	K-NN 1; don't cross window size 0
RF	Max Dept 0; number features 0; num trees 10; seed 1

			Statistic	Std. Error
Spectral Centroid	Mean		23.71	.262
	95% Confidence Interval for Mean	Lower Bound	23.19	
		Upper Bound	24.22	
	5% Trimmed Mean		21.81	
	Median		19.08	
	Variance		344.335	
	Std. Deviation		18.556	
	Minimum		1	
	Maximum		191	
	Range		190	
	Interquartile Range		21	
	Skewness		2.132	.035
	Kurtosis		8.074	.069
Spectral Rolloff	Mean		.16	.002
	95% Confidence Interval for Mean	Lower Bound	.16	
		Upper Bound	.17	
	5% Trimmed Mean		.15	
	Median		.11	
	Variance		.024	
	Std. Deviation		.155	
	Minimum		0	
	Maximum		1	
	Range		1	
	Interquartile Range		0	
	Skewness		1.762	.035
	Kurtosis		3.964	.069
Spectral Energy	Mean		.13	.001
	95% Confidence Interval for Mean	Lower Bound	.12	
		Upper Bound	.13	
	5% Trimmed Mean		.12	
	Median		.11	
	Variance		.008	
	Std. Deviation		.090	
	Minimum		0	
	Maximum		1	
	Kurtosis		3.964	.069
Spectral Energy	Mean		.13	.001
	95% Confidence Interval for Mean	Lower Bound	.12	
		Upper Bound	.13	

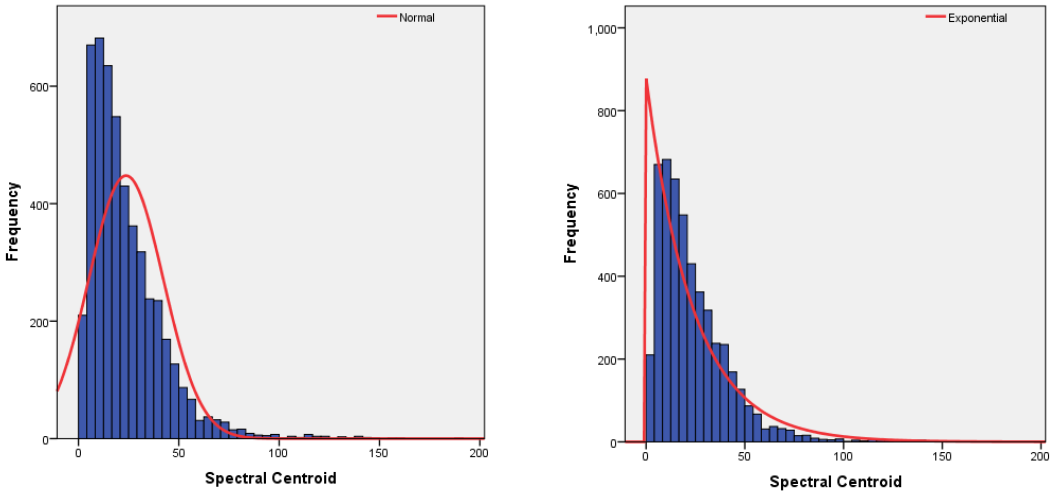
Figure 35: Description after sample reduction.



(a) ECDF for Centroid

(b) ECDF for Rolloff

Figure 36: Empirical cumulative distribution functions for centroid and rolloff.



(a) Normal Distribution on Spectral Centroid

(b) Exponentially Decreasing Distribution on Spectral Centroid

Figure 37: Other examples of probability distributions on spectral centroid.

Table 29: Suggested features to discriminate a particular genre.

Blues	Compactness Strongest Beat Chroma OBSIR
Classical	Variability Compactness Decrease Onset etection
Country	Chroma OBSIR Crest Factor MFCC
Disco	Crest factor Spectral Rolloff Strongest Beat Zero-crossing
Hiphop	Compactness Spectral Centroid Energy Strongest Beat
Jazz	Compactness Onset etection Variability Energy
Metal	Loudness Peak-based features Spectral Centroid Energy
Pop	Onset Detection Variability Spectral Rolloff Energy
Reggae	Strongest Beat Chroma OBSIR Flattness
Rock	Total Loudness MFCCs OBSIR LPC & LSF

C

CLASSIFICATION ALGORITHMS

C.0.1 *Support Vector Machines*

The aim of the support vector machine is to find the optimal equation of the hyperplane that best classifies two datasets. This optimal hyperplane maximises the margin between the boundary points of the two datasets.

Definition C.1. *Hyperplane*
Let $a_1, a_2, a_3, ..., a_n$ be scalars where $a_i \in \mathbb{R}$ and $a_i \neq 0$ for i in$(1, ..., n)$. Then the set H of all vectors

$$X = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}, \tag{50}$$

in $\mathbb{R}^n$ such that

$$a_1x_1 + a_2x_2 + a_3x_3 + ... + a_nx_n = c, \tag{51}$$

for c is a constant is a subspace of $\mathbb{R}^n$ called a hyperplane.

The *margin* is defined as the width of the line parallel to the hyperplane such that there are no points (from either set) between the hyperplane and the margin. Two such margins will exist on both sides of the hyperplane. Figure 38 shows a SVM classification of two datasets: ψ and ϕ. ψ_1^*, ψ_2^* , ϕ_{11}^*, ϕ_{16}^* are called support-vectors as they are the closest points to the separating hyperplane and therefore lie on the margin boundaries.

Definition C.2. *Support Vector Machine*
Let $X = \{x_1, x_2, x_3, ..., x_n\}$ for $X \in \mathbb{R}^m$ be a set of points and let Y define two categories

$$Y = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \tag{52}$$

then the equation of the hyperplane is

$$< W, X > + c \tag{53}$$

where $W \in \mathbb{R}^m$, $< W, X >$ is the dot product and $c \in \mathbb{R}$.

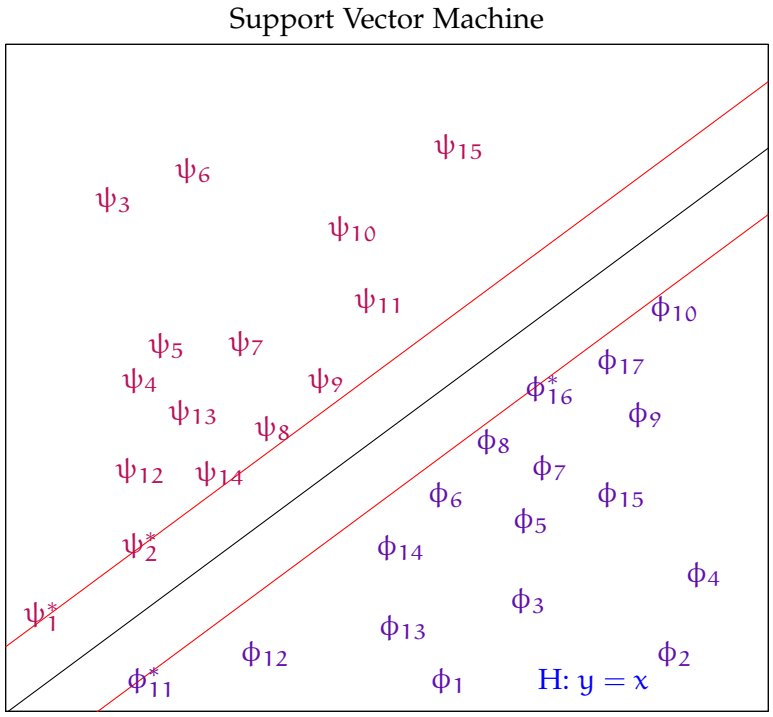


Figure 38: Classification of two datasets using a support vector machine.

Definition C.3. *Optimising the Hyperplane*
The best separating hyperplane is defined when W and c minimise ||W|| for (X,Y) such that

$$Y(< W, X > +c) \geqslant 1 \tag{54}$$

The support vectors are the X values which appear on the boundary i.e. Y(<W,X> + c) = 1.

C.0.2 *Naïve Bayes Classifier*

C.0.2.1 *Introduction*

When features are independent from one another within each class, the *Naïve Bayes classifier* is often used. Although this assumption is usually made on the feature set, the algorithm has appeared to work even when this assumption is not valid. The classification process for the *Naïve Bayes classifier* is as follows [Krauss *et al.* 1994]:

1. **Training step:** "Using the training samples, the method estimates the parameters of a probability distribution, assuming features are conditionally independent given the class."
2. **Prediction step:** "For any unseen test sample, the method computes the posterior probability of that sample belonging to each class. The method then classifies the test sample according the largest posterior probability."

C.0.2.2 *The Naïve Bayes Classification Process*

Given a DTS d and a class c, if one wants to calculate the conditional probability of a DTS d belonging to a class c this can be done by using Bayes rule which is equal to the probability of the DTS given a class multiplied by the probability of the class over the probability of the DTS:

$$P(c|d) = \frac{P(d|c)P(c)}{P(d)} \tag{55}$$

Equation 55 can be used to select the most appropriate class for DTS d. The best class is, out of all classes, is the one that maximises the probability of the class given the DTS. Therefore, if one is looking for the class where the DTS is the greatest, by Bayes rule, that should be the same as whichever class maximises the probability of c given d:

$$C_{MAP} = \underset{c \in C}{\operatorname{argmax}}\, P(c|d), \tag{56}$$

also maximises the probability of d given c multiplied by the probability of c:

$$C_{MAP} = \underset{c \in C}{\operatorname{argmax}}\, \frac{P(d|c)P(c)}{P(d)}. \tag{57}$$

As is traditional in Bayesian classification, whichever class maximises Equation 57 also maximises:

$$C_{MAP} = \underset{c \in C}{\operatorname{argmax}} P(d|c)P(c), \tag{58}$$

where the two qualities $P(d|c)$ and $P(c)$ are learnt from the data during training. This is done by dropping the denominator as the probability of the DTS will remain the same for every class - this makes the probability of the DTS a constant which is also expensive to compute. Therefore the most likely class will be the one that maximises the product of the two probabilities given by Equation 58. Each DTS can be measured by different features, therefore, $P(d|c)$ can be seen as the probability of a vector of features given a class:

$$C_{MAP} = \underset{c \in C}{\operatorname{argmax}} P(x_1, x_2, \ldots, x_n|c)P(c) \tag{59}$$

To calculate the probability of the class one would compute how often that class occurs. Since the feature set might be large the complexity of $O(|X|^n.|C|)$ is expected, which is quite large, therefore, assumptions must be met to reduce this complexity. The following assumptions can be made:

- *Bag of Words assumption*: Assuming that the feature position is negligible,
- and assume that *all feature probabilities*, $P(x_i|c_j)$, are *independent*.

Therefore using these assumptions one can represent the probability as:

$$P(x_1, ..., x_n|c) = P(x_1|c) \cdot P(x_2|c) \cdot \ldots \cdot p(x_n|c). \tag{60}$$

That is the joint probability of every feature conditioned on a class as the product of a whole collection of independent probabilities as shown in Equation 60. In other words in order to compute the simplifying naïve Bayes assumption to compute the most likely class by the multiplying a likelihood of a set of features times the probility of a class can be simplified as the *best class*, by the naïve Bayes assumption, is the class that maximises the prior probability of the class multiplied by (for every feature in a set of features) the probability of that feature given the class:

$$c_{NB} = \underset{c_j \in C}{\operatorname{argmax}} \prod_{i \in \text{POSITIONS}} P(x_i|c_j) \tag{61}$$

C.0.3 *K - Nearest Neighbours*

The K-nearest neighbours classification model is used to map new points to a class based on a given training set. The nearest neighbours algorithm finds the k nearest neighbours in the feature space of a point and uses this information to classify the point into one of n classes [Altman 1992]. The nearest neighbours technique only approximates points using a local setting. Furthermore, most of the computation is left until the classification is actually necessary, and so this method is considered the simplest classification method from all classification techniques available. Although the neighbours technique is considered simple, it is known also for its usefulness

when considering the contribution of a new point's neighbours' weights, making its classification decisions mostly on the closest points rather then considering all of the points fairly. For example, if one would want to classify a new point p, then a weighting problem is considered where all neighbouring points are given $\frac{1}{s_i}$, where s_i is the distance from p to the i^{th} neighbour. When using the nearest neighbours method for classification the closest points from p are taken from a set of objects with a known class. A big disadvantage is that the method is sensitive to local structure. Altman [1992] have shown that the nearest neighbours technique can be used for regression as well. Haggblade *et al.* [2011] used k-nearest neighbours to classify music genre.

BIBLIOGRAPHY

[Adams *et al.* 2004] Norman H Adams, Mark A Bartsch, Jonah B Shifrin, and Gregory H Wakefield. Time series alignment for music information retrieval. *Ann Arbor*, 1001:48109–2110, 2004.

[Aha and Kibler 1991] D. Aha and D. Kibler. Instance-based learning algorithms. *Machine Learning*, 6:37–66, 1991.

[Ahumada Jr 1996] AJ Ahumada Jr. Perceptual classification images from vernier acuity masked by noise. *Perception*, 26(Suppl 18):18, 1996.

[Ajoodha *et al.* 2014] Ritesh Ajoodha, Richard Klein, and Marija Jakovljevic. Using statistical models and evolutionary algorithms in algorithmic music composition. In Khosrow-Pour Mehdi, editor, *The Encyclopedia of Information Science and Technology*. IGI Global, Hershey, Pennsylvania, United States, 3rd edition, 2014.

[Alexandre-Cortizo *et al.* 2005] Enrique Alexandre-Cortizo, Manuel Rosa-Zurera, and Francisco Lopez-Ferreras. Application of fisher linear discriminant analysis to speech/music classification. In *Computer as a Tool, 2005. EUROCON 2005. The International Conference on*, volume 2, pages 1666–1669. IEEE, 2005.

[Altman 1992] Naomi S Altman. An introduction to kernel and nearest-neighbor nonparametric regression. *The American Statistician*, 46(3):175–185, 1992.

[Angeles 2009] Bruno Angeles. Metadata and jmusicmetamanager. 2009.

[Aucouturier and Pachet 2002] Jean-Julien Aucouturier and Francois Pachet. Music similarity measures: What's the use? In *ISMIR*, 2002.

[Aucouturier and Pachet 2003] Jean-Julien Aucouturier and Francois Pachet. Representing musical genre: A state of the art. *Journal of New Music Research*, 32(1):83–93, 2003.

[Bäckström and Magi 2006] Tom Bäckström and Carlo Magi. Properties of line spectrum pair polynomials-a review. *Signal processing*, 86(11):3286–3298, 2006.

[bal] *Ball Room dances dataset*. http://www.ballroomdancers.com. Accessed: 2014-07-10.

[Basili *et al.* 2004] Roberto Basili, Alfredo Serafini, and Armando Stellato. Classification of musical genre: a machine learning approach. In *ISMIR*. Citeseer, 2004.

[Benetos and Kotropoulos 2008] Emmanouil Benetos and Constantine Kotropoulos. A tensor-based approach for automatic music genre classification. In *Proceedings of the European Signal Processing Conference*, 2008.

[Berenzweig *et al.* 2003] Adam Berenzweig, Daniel PW Ellis, and Steve Lawrence. Anchor space for classification and similarity measurement of music. In *Multimedia and Expo, 2003. ICME'03. Proceedings. 2003 International Conference on*, volume 1, pages I–29. IEEE, 2003.

[Berenzweig *et al.* 2004] Adam Berenzweig, Beth Logan, Daniel PW Ellis, and Brian Whitman. A large-scale evaluation of acoustic and subjective music-similarity measures. *Computer Music Journal*, 28(2):63–76, 2004.

[Bergroth *et al.* 2000] Lasse Bergroth, Harri Hakonen, and Timo Raita. A survey of longest common subsequence algorithms. In *String Processing and Information Retrieval, 2000. SPIRE 2000. Proceedings. Seventh International Symposium on*, pages 39–48. IEEE, 2000.

[Bergstra *et al.* 2006] James Bergstra, Norman Casagrande, Dumitru Erhan, Douglas Eck, and Balázs Kégl. Aggregate features and adaboost for music classification. *Machine Learning*, 65(2-3):473–484, 2006.

[Brandt *et al.* 2012] Anthony Brandt, Molly Gebrian, and L Robert Slevc. Music and early language acquisition. *Frontiers in psychology*, 3, 2012.

[Breiman 2001] Leo Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.

[Cartwright 2007] Kenneth V Cartwright. Determining the effective or rms voltage of various waveforms without calculus. *Technology Interface*, 8(1):20, 2007.

[Casey *et al.* 2008] Michael A Casey, Remco Veltkamp, Masataka Goto, Marc Leman, Christophe Rhodes, and Malcolm Slaney. Content-based music information retrieval: Current directions and future challenges. *Proceedings of the IEEE*, 96(4):668–696, 2008.

[Cast *et al.* 2014] John Cast, Chris Schulze, and Ali Fauci. Music genre classification. 2014.

[Cataltepe *et al.* 2007] Zehra Cataltepe, Yusuf Yaslan, and Abdullah Sonmez. Music genre classification using midi and audio features. *EURASIP Journal on Applied Signal Processing*, 2007(1):150–150, 2007.

[Chang and Lin 2001] Chih-Chung Chang and Chih-Jen Lin. *LIBSVM - A Library for Support Vector Machines*, 2001. The Weka classifier works with version 2.82 of LIBSVM.

[Chen 1988] Chi-hau Chen. *Signal processing handbook*, volume 51. CRC Press, 1988.

[Cilibrasi *et al.* 2004] Rudi Cilibrasi, Paul Vitányi, and Ronald De Wolf. Algorithmic clustering of music based on string compression. *Computer Music Journal*, 28(4):49–67, 2004.

[Cohen and Lefebvre 2005] Henri Cohen and Claire Lefebvre. *Handbook of categorization in cognitive science*. Elsevier, 2005.

[Cohen 1995] William W. Cohen. Fast effective rule induction. In *Twelfth International Conference on Machine Learning*, pages 115–123. Morgan Kaufmann, 1995.

[Cox *et al.* 1997] Ingemar J Cox, Joe Kilian, F Thomson Leighton, and Talal Shamoon. Secure spread spectrum watermarking for multimedia. *Image Processing, IEEE Transactions on*, 6(12):1673–1687, 1997.

[Crammer and Singer 2002] Koby Crammer and Yoram Singer. On the algorithmic implementation of multiclass kernel-based vector machines. *The Journal of Machine Learning Research*, 2:265–292, 2002.

[Dannenberg *et al.* 1997] Roger B Dannenberg, Belinda Thom, and David Watson. A machine learning approach to musical style recognition. 1997.

[David 2000] Pye David. Content-based methods for managing electronic music. In *International Conference on Acoustic Speech and Signal Processing*. IEEE, 2000.

[Davis and Mermelstein 1980] Steven Davis and Paul Mermelstein. Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. *Acoustics, Speech and Signal Processing, IEEE Transactions on*, 28(4):357–366, 1980.

[Deshpande *et al.* 2001] Hrishikesh Deshpande, Rohit Singh, and Unjung Nam. Classification of music signals in the visual domain. In *Proceedings of the COST-G6 Conference on Digital Audio Effects*, pages 1–4. sn, 2001.

[Dictionary 1971] Oxford English Dictionary. Compact edition. *Volume Two*, 130, 1971.

[Dietterich and Bakiri 1995] Thomas G. Dietterich and Ghulum Bakiri. Solving multiclass learning problems via error-correcting output codes. *arXiv preprint cs/9501101*, 1995.

[Dixon *et al.* 2004] Simon Dixon, Fabien Gouyon, and Gerhard Widmer. Towards characterisation of music via rhythmic patterns. In *ISMIR*, 2004.

[Dixon 2006] Simon Dixon. Onset detection revisited. In *Proceedings of the 9th International Conference on Digital Audio Effects*, volume 120, pages 133–137. Citeseer, 2006.

[Downie 2003] J Stephen Downie. Toward the scientific evaluation of music information retrieval systems. In *ISMIR*, 2003.

[Duan and Keerthi 2005] Kai-Bo Duan and S Sathiya Keerthi. Which is the best multiclass svm method? an empirical study. In *Multiple Classifier Systems*, pages 278–285. Springer, 2005.

[Dubnov 2004] Shlomo Dubnov. Generalization of spectral flatness measure for non-gaussian linear processes. *Signal Processing Letters, IEEE*, 11(8):698–701, 2004.

[Dunn 2010] Patrick F Dunn. *Measurement and data analysis for engineering and science*. CRC Press, 2010.

[Duxbury *et al.* 2003] Chris Duxbury, Juan Pablo Bello, Mike Davies, Mark Sandler, et al. Complex domain onset detection for musical signals. In *Proc. Digital Audio Effects Workshop (DAFx)*, number 1, pages 6–9, 2003.

[EL-Manzalawy 2005] Yasser EL-Manzalawy. *WLSVM*, 2005. You don't need to include the WLSVM package in the CLASSPATH.

[Ellis and Poliner 2007] Daniel PW Ellis and Graham E Poliner. Identifyingcover songs' with chroma features and dynamic programming beat tracking. In *Acoustics, Speech and Signal Processing, 2007. ICASSP 2007. IEEE International Conference on*, volume 4, pages IV–1429. IEEE, 2007.

[Ellis *et al.*] Dan Ellis, Adam Berenzweig, and Brian Whitman. *The "uspop2002" pop music data set*. http://labrosa.ee.columbia.edu/projects/musicsim/uspop2002.html. Accessed: 2005-07-09.

[Ellis 2007] Daniel PW Ellis. Classifying music audio with timbral and chroma features. In *ISMIR 2007: Proceedings of the 8th International Conference on Music Information Retrieval: September 23-27, 2007, Vienna, Austria*, pages 339–340. Austrian Computer Society, 2007.

[Eronen and Antti 2001] Eronen and Antti. Automatic musical instrument recognition. *Master's thesis, Tampere university of technology*, 2001.

[Essid *et al.* 2006] Slim Essid, Gaël Richard, and Bertrand David. Instrument recognition in polyphonic music based on automatic taxonomies. *Audio, Speech, and Language Processing, IEEE Transactions on*, 14(1):68–80, 2006.

[Essid 2005] Slim Essid. *Classification automatique des signaux audio-fréquences: reconnaissance des instruments de musique*. PhD thesis, Université Pierre et Marie Curie-Paris VI, 2005.

[Foote and Uchihashi 2001] Jonathan Foote and Shingo Uchihashi. The beat spectrum: A new approach to rhythm analysis. In *ICME*, 2001.

[Foote 1997] Jonathan T Foote. Content-based retrieval of music and audio. In *Voice, Video, and Data Communications*, pages 138–147. International Society for Optics and Photonics, 1997.

[Fry 1979] Dennis Butler Fry. *The physics of speech*. Cambridge University Press, 1979.

[Fu *et al.* 2011] Zhouyu Fu, Guojun Lu, Kai Ming Ting, and Dengsheng Zhang. A survey of audio-based music classification and annotation. *Multimedia, IEEE Transactions on*, 13(2):303–319, 2011.

[Fujinaga 1996] Ichiro Fujinaga. *Adaptive optical music recognition*. PhD thesis, McGill University, 1996.

[Furui 1986] Sadaoki Furui. Speaker-independent isolated word recognition using dynamic features of speech spectrum. *Acoustics, Speech and Signal Processing, IEEE Transactions on*, 34(1):52–59, 1986.

[Giannoulis *et al.* 2012] Dimitrios Giannoulis, Michael Massberg, and Joshua D Reiss. Digital dynamic range compressor design-a tutorial and analysis. *Journal of the Audio Engineering Society*, 60(6):399–408, 2012.

[Gjerdingen and Perrott 2008] Robert O Gjerdingen and David Perrott. Scanning the dial: The rapid recognition of music genres. *Journal of New Music Research*, 37(2):93–100, 2008.

[Goto and Muraoka 1994] Masataka Goto and Yoichi Muraoka. A beat tracking system for acoustic signals of music. In *Proceedings of the second ACM international conference on Multimedia*, pages 365–372. ACM, 1994.

[Gouyon *et al.* 2000] Fabien Gouyon, François Pachet, and Olivier Delerue. On the use of zero-crossing rate for an application of classification of percussive sounds. In *Proceedings of the COST G-6 conference on Digital Audio Effects (DAFX-00), Verona, Italy*. Citeseer, 2000.

[Gouyou *et al.* 2000] F Gouyou, F Pachet, and O Delerue. Classifying percussive sounds: a matter of zero-crossing rate? In *Proceedings of the COST G-6 Conference on Digital Audio Effects*, 2000.

[Grey and Gordon 1978] John M Grey and John W Gordon. Perceptual effects of spectral modifications on musical timbres. *The Journal of the Acoustical Society of America*, 63(5):1493–1500, 1978.

[Grimaldi *et al.* 2003] Marco Grimaldi, Pádraig Cunningham, and Anil Kokaram. A wavelet packet representation of audio signals for music genre classification using different ensemble and feature selection techniques. In *Proceedings of the 5th ACM SIGMM international workshop on Multimedia information retrieval*, pages 102–108. ACM, 2003.

[Guo and Li 2003] Guodong Guo and Stan Z Li. Content-based audio classification and retrieval by support vector machines. *Neural Networks, IEEE Transactions on*, 14(1):209–215, 2003.

[Haggblade *et al.* 2011] Michael Haggblade, Yang Hong, and Kenny Kao. Music genre classification. *Department of Computer Science, Stanford University*, 2011.

[Hamel *et al.* 2011] Philippe Hamel, Simon Lemieux, Yoshua Bengio, and Douglas Eck. Temporal pooling and multiscale learning for automatic annotation and ranking of music audio. In *ISMIR*, pages 729–734, 2011.

[Helen and Virtanen 2005] Marko Helen and Tuomas Virtanen. Separation of drums from polyphonic music using non-negative matrix factorization and support vector machine. In *Proc. EUSIPCO*, volume 2005, 2005.

[Herre *et al.* 2001] A Herre, Eric Allamanche, and Oliver Hellmuth. Robust matching of audio signals using spectral flatness features. In *Applications of Signal Processing to Audio and Acoustics, 2001 IEEE Workshop on the*, pages 127–130. IEEE, 2001.

[Holzapfel and Stylianou 2008] André Holzapfel and Yannis Stylianou. Musical genre classification using nonnegative matrix factorization-based features. *Audio, Speech, and Language Processing, IEEE Transactions on*, 16(2):424–434, 2008.

[Hsu and Lin 2002] Chih-Wei Hsu and Chih-Jen Lin. A comparison of methods for multiclass support vector machines. *Neural Networks, IEEE Transactions on*, 13(2):415–425, 2002.

[Hsue and Soliman 1990] S-Z Hsue and Samir S Soliman. Automatic modulation classification using zero crossing. In *IEE Proceedings F (Radar and Signal Processing)*, volume 137, pages 459–464. IET, 1990.

[Hu *et al.* 2002] Ning Hu, Roger B Dannenberg, and Ann L Lewis. A probabilistic model of melodic similarity. 2002.

[Iliopoulos and Kurokawa 2002] Costas S Iliopoulos and Masahiro Kurokawa. String matching with gaps for musical melodic recognition. In *Stringology*, pages 55–64, 2002.

[Iverson and Krumhansl 1993] Paul Iverson and Carol L Krumhansl. Isolating the dynamic attributes of musical timbrea). *The Journal of the Acoustical Society of America*, 94(5):2595–2603, 1993.

[Jacobs 1994] Harold R Jacobs. *Mathematics: A human endeavor*. Macmillan, 1994.

[Jang and Gao 2000] J-S Roger Jang and Ming-Yang Gao. A query-by-singing system based on dynamic programming. In *International Workshop on Intelligent Systms Resolutions (the 8th Bellman Continuum)*, pages 85–89, 2000.

[Jang *et al.* 2008] Dalwon Jang, Minho Jin, and Chang Dong Yoo. Music genre classification using novel features and a weighted voting method. In *Multimedia and Expo, 2008 IEEE International Conference on*, pages 1377–1380. IEEE, 2008.

[Jiang *et al.* 2002] Dan-Ning Jiang, Lie Lu, Hong-Jiang Zhang, Jian-Hua Tao, and Lian-Hong Cai. Music type classification by spectral contrast feature. In *Multimedia and Expo, 2002. ICME'02. Proceedings. 2002 IEEE International Conference on*, volume 1, pages 113–116. IEEE, 2002.

[John and Langley 1995] George H. John and Pat Langley. Estimating continuous distributions in bayesian classifiers. In *Eleventh Conference on Uncertainty in Artificial Intelligence*, pages 338–345, San Mateo, 1995. Morgan Kaufmann.

[Kankanhalli and Ramakrishnan 1998] Mohan S Kankanhalli and KR Ramakrishnan. Content based watermarking of images. In *Proceedings of the sixth ACM international conference on Multimedia*, pages 61–70. ACM, 1998.

[Khadkevich and Omologo 2009] Maksim Khadkevich and Maurizio Omologo. Use of hidden markov models and factored language models for automatic chord recognition. In *ISMIR*, pages 561–566, 2009.

[Kim *et al.* 2004] Hyoung-Gook Kim, Nicolas Moreau, and Thomas Sikora. Audio classification based on mpeg-7 spectral basis representations. *Circuits and Systems for Video Technology, IEEE Transactions on*, 14(5):716–725, 2004.

[Klapuri and Davy 2006] Anssi Klapuri and Manuel Davy. *Signal processing methods for music transcription*, volume 1. Springer, 2006.

[Kotsifakos *et al.* 2011] Alexios Kotsifakos, Panagiotis Papapetrou, Jaakko Hollmén, and Dimitrios Gunopulos. A subsequence matching with gaps-range-tolerances framework: a query-by-humming application. *Proceedings of the VLDB Endowment*, 4(11):761–771, 2011.

[Kotsifakos *et al.* 2012] Alexios Kotsifakos, Panagiotis Papapetrou, Jaakko Hollmén, Dimitrios Gunopulos, and Vassilis Athitsos. A survey of query-by-humming similarity methods. In *Proceedings of the 5th International Conference on PErvasive Technologies Related to Assistive Environments*, page 5. ACM, 2012.

[Krauss *et al.* 1994] Thomas P Krauss, Loren Shure, and John Little. *Signal processing toolbox for use with MATLAB®: user's guide*. The MathWorks, 1994.

[Lambrou *et al.* 1998] Tryphon Lambrou, PSRSM Kudumakis, R Speller, M Sandler, and A Linney. Classification of audio signals using statistical features on time and wavelet transform domains. In *Acoustics, Speech and Signal Processing, 1998.*

Proceedings of the 1998 IEEE International Conference on, volume 6, pages 3621–3624. IEEE, 1998.

[Landwehr *et al.* 2005] Niels Landwehr, Mark Hall, and Eibe Frank. Logistic model trees. 95(1-2):161–205, 2005.

[Laroche 2001] Jean Laroche. Estimating tempo, swing and beat locations in audio recordings. In *Applications of Signal Processing to Audio and Acoustics, 2001 IEEE Workshop on the*, pages 135–138. IEEE, 2001.

[Lee and Downie 2004] Jin Ha Lee and J Stephen Downie. Survey of music information needs, uses, and seeking behaviours: Preliminary findings. In *ISMIR*, volume 2004, page 5th, 2004.

[Lee *et al.* 2001] Yoonkyung Lee, Yi Lin, and Grace Wahba. Multicategory support vector machines. In *Proceedings of the 33rd Symposium on the Interface*. Citeseer, 2001.

[Lee *et al.* 2004] Yoonkyung Lee, Yi Lin, and Grace Wahba. Multicategory support vector machines: Theory and application to the classification of microarray data and satellite radiance data. *Journal of the American Statistical Association*, 99(465):67–81, 2004.

[Lee *et al.* 2007] Chang-Hsing Lee, Jau-Ling Shih, Kun-Ming Yu, and Jung-Mau Su. Automatic music genre classification using modulation spectral contrast feature. In *ICME*, pages 204–207, 2007.

[Lemström and Ukkonen 2000] Kjell Lemström and Esko Ukkonen. Including interval encoding into edit distance based music comparison and retrieval. In *Proc. AISB*, pages 53–60, 2000.

[Li and Ogihara 2006] Tao Li and Mitsunori Ogihara. Toward intelligent music information retrieval. *Multimedia, IEEE Transactions on*, 8(3):564–574, 2006.

[Li *et al.* 2001] Dongge Li, Ishwar K Sethi, Nevenka Dimitrova, and Tom McGee. Classification of general audio data for content-based retrieval. *Pattern recognition letters*, 22(5):533–544, 2001.

[Li *et al.* 2003] Tao Li, Mitsunori Ogihara, and Qi Li. A comparative study on content-based music genre classification. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, pages 282–289. ACM, 2003.

[Li 2000] Stan Z Li. Content-based audio classification and retrieval using the nearest feature line method. *Speech and Audio Processing, IEEE Transactions on*, 8(5):619–625, 2000.

[Lidy and Rauber 2005] Thomas Lidy and Andreas Rauber. Evaluation of feature extractors and psycho-acoustic transformations for music genre classification. In *ISMIR*, pages 34–41, 2005.

[Lidy *et al.* 2007] T Lidy, A Rauber, A Pertusa, and J Inesta. Combining audio and symbolic descriptors for music classification from audio. *Music Information Retrieval Information Exchange (MIREX)*, 2007.

[Lippens *et al.* 2004] Stefaan Lippens, Jean-Pierre Martens, and Tom De Mulder. A comparison of human and automatic musical genre classification. In *Acoustics, Speech, and Signal Processing, 2004. Proceedings.(ICASSP'04). IEEE International Conference on*, volume 4, pages iv–233. IEEE, 2004.

[Liu and Huang 2000] Zhu Liu and Qian Huang. Content-based indexing and retrieval-by-example in audio. In *Multimedia and Expo, 2000. ICME 2000. 2000 IEEE International Conference on*, volume 2, pages 877–880. IEEE, 2000.

[Liu 1999] Samson J Liu. *Perceptual image resolution enhancement system*, March 9 1999. US Patent 5,880,767.

[Logan and Salomon 2001] Beth Logan and Ariel Salomon. *Music similarity function based on signal analysis*, 2001.

[Logan 2000] Beth Logan. Mel frequency cepstral coefficients for music modeling. In *ISMIR*, 2000.

[Lu *et al.* 2002] Lie Lu, Hong-Jiang Zhang, and Hao Jiang. Content analysis for audio classification and segmentation. *Speech and Audio Processing, IEEE Transactions on*, 10(7):504–516, 2002.

[MAG] *Magnatagatune Music Database*. http://tagatune.org/Magnatagatune.html. Accessed: 2014-07-09.

[Makhoul 1975] John Makhoul. Linear prediction: A tutorial review. *Proceedings of the IEEE*, 63(4):561–580, 1975.

[Malm *et al.* 1967] William P Malm, Christopher B Field, and MR Raupach. *Music cultures of the Pacific, the Near East, and Asia*. Prentice-Hall, 1967.

[Mandel and Ellis 2005] Michael I Mandel and Daniel PW Ellis. Song-level features and support vector machines for music classification. In *ISMIR 2005: 6th International Conference on Music Information Retrieval: Proceedings: Variation 2: Queen Mary, University of London & Goldsmiths College, University of London, 11-15 September, 2005*, pages 594–599. Queen Mary, University of London, 2005.

[Mandel and Ellis 2007] M Mandel and D Ellis. Labrosa's audio music similarity and classification submissions. *Music Information Retrieval Information Exchange (MIREX)*, 2007.

[Martin 1999] Keith Dana Martin. *Sound-source recognition: A theory and computational model*. PhD thesis, Massachusetts Institute of Technology, 1999.

[Mathews and Walker 1970] Jon Mathews and Robert Lee Walker. *Mathematical methods of physics*, volume 271. WA Benjamin New York, 1970.

[Mathieu *et al.* 2010] Benoit Mathieu, Slim Essid, Thomas Fillon, Jacques Prado, and Gaël Richard. Yaafe, an easy to use and efficient audio feature extraction software. In *ISMIR*, pages 441–446, 2010.

[Mazzoni and Dannenberg 2001] Dominic Mazzoni and Roger B Dannenberg. Melody matching directly from audio. In *2nd Annual International Symposium on Music Information Retrieval*, pages 17–18. Citeseer, 2001.

[McFee *et al.* 2012] Brian McFee, Luke Barrington, and Gert Lanckriet. Learning content similarity for music recommendation. *Audio, Speech, and Language Processing, IEEE Transactions on*, 20(8):2207–2218, 2012.

[McKay and Fujinaga 2004] Cory McKay and Ichiro Fujinaga. Automatic genre classification using large high-level musical feature sets. In *ISMIR*, volume 2004, pages 525–530, 2004.

[McKay and Fujinaga 2005] Cory McKay and Ichiro Fujinaga. Automatic music classification and the importance of instrument identification. In *Proceedings of the Conference on Interdisciplinary Musicology*, 2005.

[McKay and Fujinaga 2006] Cory McKay and Ichiro Fujinaga. Musical genre classification: Is it worth pursuing and how can it be improved? In *ISMIR*, pages 101–106, 2006.

[McKay *et al.* 2005a] Cory McKay, Rebecca Fiebrink, Daniel McEnnis, Beinan Li, and Ichiro Fujinaga. Ace: A framework for optimizing music classification. In *ISMIR*, pages 42–49, 2005.

[McKay *et al.* 2005b] Cory McKay, Ichiro Fujinaga, and Philippe Depalle. jaudio: A feature extraction library. In *Proceedings of the International Conference on Music Information Retrieval*, pages 600–3, 2005.

[McKay 2004] Cory McKay. *Automatic genre classification of MIDI recordings*. PhD thesis, McGill University, 2004.

[McKinney and Breebaart 2003] Martin F McKinney and Jeroen Breebaart. Features for audio and music classification. In *ISMIR*, volume 3, pages 151–158, 2003.

[McNab *et al.* 1996] Rodger J McNab, Lloyd A Smith, Ian H Witten, Clare L Henderson, and Sally Jo Cunningham. Towards the digital music library: Tune retrieval from acoustic input. In *Proceedings of the first ACM international conference on Digital libraries*, pages 11–18. ACM, 1996.

[Medhi 1992] Jyotiprasad Medhi. *Statistical methods: an introductory text*. New Age International, 1992.

[Meek and Birmingham 2004] Colin Meek and William P Birmingham. A comprehensive trainable error model for sung music queries. *J. Artif. Intell. Res.(JAIR)*, 22:57–91, 2004.

[Meng *et al.* 2007] Anders Meng, Peter Ahrendt, Jan Larsen, and Lars Kai Hansen. Temporal feature integration for music genre classification. *Audio, Speech, and Language Processing, IEEE Transactions on*, 15(5):1654–1664, 2007.

[Mongeau and Sankoff 1990] Marcel Mongeau and David Sankoff. Comparison of musical sequences. *Computers and the Humanities*, 24(3):161–175, 1990.

[Moore *et al.* 1997] Brian CJ Moore, Brian R Glasberg, and Thomas Baer. A model for the prediction of thresholds, loudness, and partial loudness. *Journal of the Audio Engineering Society*, 45(4):224–240, 1997.

[Narvekar and Karam 2009] Niranjan D Narvekar and Lina J Karam. A no-reference perceptual image sharpness metric based on a cumulative probability of blur detection. In *Quality of Multimedia Experience, 2009. QoMEx 2009. International Workshop on*, pages 87–91. IEEE, 2009.

[Nelwamondo *et al.* 2006] Fulufhelo V Nelwamondo, Tshilidzi Marwala, and Unathi Mahola. Early classifications of bearing faults using hidden markov models, gaussian mixture models, mel-frequency cepstral coefficients and fractals. *International Journal of Innovative Computing, Information and Control*, 2(6):1281–1299, 2006.

[Noll and Schroeder 2005] A Michael Noll and MR Schroeder. Short-time "cepstrum" pitch detection. *The Journal of the Acoustical Society of America*, 36(5):1030–1030, 2005.

[Noll 2005a] A Michael Noll. Cepstrum pitch determination. *The journal of the acoustical society of America*, 41(2):293–309, 2005.

[Noll 2005b] A Michael Noll. Short-time spectrum and "cepstrum" techniques for vocal-pitch detection. *The Journal of the Acoustical Society of America*, 36(2):296–302, 2005.

[North and Hargreaves 1997] Adrian C North and David J Hargreaves. Liking for musical styles. *Musicae Scientiae*, 1(1):109–128, 1997.

[Owen 2000] Harold Owen. *Music theory resource book*. Oxford University Press, 2000.

[Oxford 1989] English Dictionary Oxford. *Oxford: Oxford University Press*, 1989.

[Pachet and Aucouturier 2004] Francois Pachet and Jean-Julien Aucouturier. Improving timbre similarity: How high is the sky? *Journal of negative results in speech and audio sciences*, 1(1):1–13, 2004.

[Pampalk *et al.* 2003] Elias Pampalk, Simon Dixon, and Gerhard Widmer. On the evaluation of perceptual similarity measures for music. In *of: Proceedings of the sixth international conference on digital audio effects (DAFx-03)*, pages 7–12, 2003.

[Pampalk *et al.* 2005] Elias Pampalk, Arthur Flexer, and Gerhard Widmer. Improvements of audio-based music similarity and genre classificaton. In *ISMIR*, volume 5, pages 634–637. London, UK, 2005.

[Panagakis *et al.* 2008] Ioannis Panagakis, Emmanouil Benetos, and Constantine Kotropoulos. Music genre classification: A multilinear approach. In *ISMIR*, pages 583–588, 2008.

[Panagakis *et al.* 2010] Yannis Panagakis, Constantine Kotropoulos, and Gonzalo R Arce. Non-negative multilinear principal component analysis of auditory temporal modulations for music genre classification. *Audio, Speech, and Language Processing, IEEE Transactions on*, 18(3):576–588, 2010.

[Pardo and Birmingham 2002] Bryan Pardo and William P Birmingham. Encoding timing information for musical query matching. In *ISMIR*, 2002.

[Patterson *et al.* 2010] Roy D Patterson, Etienne Gaudrain, and Thomas C Walters. The perception of family and register in musical tones. In *Music Perception*, pages 13–50. Springer, 2010.

[Pauws 2002] Steffen Pauws. Cubyhum: a fully operational" query by humming" system. In *ISMIR*. Citeseer, 2002.

[Pearce and Wiggins 2007] Marcus T Pearce and Geraint A Wiggins. Evaluating cognitive models of musical composition. In *Proceedings of the 4th international joint workshop on computational creativity*, pages 73–80. Goldsmiths, University of London, 2007.

[Peeters *et al.* 2000] Geoffroy Peeters, Stephen McAdams, and Perfecto Herrera. Instrument sound description in the context of mpeg-7. In *Proceedings of the 2000 International Computer Music Conference*, pages 166–169. Citeseer, 2000.

[Peeters 2003] Geoffroy Peeters. Automatic classification of large musical instrument databases using hierarchical classifiers with inertia ratio maximization. In *Audio Engineering Society Convention 115*. Audio Engineering Society, 2003.

[Peeters 2004] Geoffroy Peeters. {A large set of audio features for sound description (similarity and classification) in the CUIDADO project}. 2004.

[Perrot and Gjerdigen 1999] David Perrot and Robert Gjerdigen. Scanning the dial: An exploration of factors in the identification of musical style. In *Proceedings of the 1999 Society for Music Perception and Cognition*, page 88, 1999.

[Pfau *et al.* 2001] Thilo Pfau, Daniel PW Ellis, and Andreas Stolcke. Multispeaker speech activity detection for the icsi meeting recorder. In *Automatic Speech Recognition and Understanding, 2001. ASRU'01. IEEE Workshop on*, pages 107–110. IEEE, 2001.

[Plack *et al.* 2006] Christopher J Plack, Andrew J Oxenham, and Richard R Fay. *Pitch: neural coding and perception*, volume 24. Springer, 2006.

[Platt *et al.* 1999] John C Platt, Nello Cristianini, and John Shawe-Taylor. Large margin dags for multiclass classification. In *nips*, volume 12, pages 547–553, 1999.

[Pressing 2002] Jeff Pressing. Black atlantic rhythm: Its computational and transcultural foundations. *Music Perception*, 19(3):285–310, 2002.

[Quinlan 1993] Ross Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publishers, San Mateo, CA, 1993.

[Rabiner and Juang 1993] Lawrence R Rabiner and Biing-Hwang Juang. *Fundamentals of speech recognition*, volume 14. PTR Prentice Hall Englewood Cliffs, 1993.

[Rai and Kumar 2014] Anand Rai and Dharmendra Kumar. A comparative study on wrapper and filter methods for feature selection in data mining. *International Journal Of Scientific Research And Education*, 2(08), 2014.

[Randel 2003] Don Michael Randel. *The Harvard dictionary of music*, volume 16. Harvard University Press, 2003.

[Reichl and Ruske 1995] Wolfgang Reichl and Günther Ruske. A hybrid rbf-hmm system for continuous speech recognition. In *Acoustics, Speech, and Signal Processing, 1995. ICASSP-95., 1995 International Conference on*, volume 5, pages 3335–3338. IEEE, 1995.

[Reichl and Ruske 2011] Wolfgang Reichl and Günther Ruske. Discriminative training for continuous speech recognition. 2011.

[Rodet and Tisserand 2001] Xavier Rodet and Patrice Tisserand. *ECRINS: Calcul des descripteurs bas niveaux*. Technical report, Technical report, Ircam–Centre Pompidou, Paris, France, 2001.

[Roederer and Roederer 1995] Juan G Roederer and Juan G Roederer. The physics and psychophysics of music. 1995.

[Sadie 1980] Stanley E Sadie. The new grove dictionary of music and musicians. 1980.

[Sahidullah and Saha 2012] Md Sahidullah and Goutam Saha. Design, analysis and experimental evaluation of block based transformation in mfcc computation for speaker recognition. *Speech Communication*, 54(4):543–565, 2012.

[Sakurai *et al.* 2007] Yasushi Sakurai, Christos Faloutsos, and Masashi Yamamuro. Stream monitoring under the time warping distance. In *Data Engineering, 2007. ICDE 2007. IEEE 23rd International Conference on*, pages 1046–1055. IEEE, 2007.

[Saunders 1996] John Saunders. Real-time discrimination of broadcast speech/music. In *Acoustics, Speech, and Signal Processing, 1996. ICASSP-96 Vol 2. Conference Proceedings., 1996 IEEE International Conference on*, volume 2, pages 993–996. IEEE, 1996.

[Scaringella *et al.* 2006] Nicolas Scaringella, Giorgio Zoia, and Daniel Mlynek. Automatic genre classification of music content: a survey. *Signal Processing Magazine, IEEE*, 23(2):133–141, 2006.

[Scheirer and Slaney 1997] Eric Scheirer and Malcolm Slaney. Construction and evaluation of a robust multifeature speech/music discriminator. In *Acoustics, Speech, and Signal Processing, 1997. ICASSP-97., 1997 IEEE International Conference on*, volume 2, pages 1331–1334. IEEE, 1997.

[Scheirer 1998] Eric D Scheirer. Tempo and beat analysis of acoustic musical signals. *The Journal of the Acoustical Society of America*, 103(1):588–601, 1998.

[Schmidt-Jones 2011] Catherine Schmidt-Jones. Form in music. *Connexions*, 2011.

[Schubert *et al.* 2004] Emery Schubert, Joe Wolfe, and Alex Tarnopolsky. Spectral centroid and timbre in complex, multiple instrumental textures. In *Proceedings of the international conference on music perception and cognition, North Western University, Illinois*, pages 112–116, 2004.

[Schussler 1976] Hans W Schussler. A stability theorem for discrete systems. *Acoustics, Speech and Signal Processing, IEEE Transactions on*, 24(1):87–89, 1976.

[Shifrin *et al.* 2002] Jonah Shifrin, Bryan Pardo, Colin Meek, and William Birmingham. Hmm-based musical query retrieval. In *Proceedings of the 2nd ACM/IEEE-CS joint conference on Digital libraries*, pages 295–300. ACM, 2002.

[Soltau *et al.* 1998] Hagen Soltau, Tanja Schultz, Martin Westphal, and Alex Waibel. Recognition of music types. In *Acoustics, Speech and Signal Processing, 1998. Proceedings of the 1998 IEEE International Conference on*, volume 2, pages 1137–1140. IEEE, 1998.

[Song and Zhang 2008] Yangqiu Song and Changshui Zhang. Content-based information fusion for semi-supervised music genre classification. *Multimedia, IEEE Transactions on*, 10(1):145–152, 2008.

[Sturm 2012a] Bob L Sturm. An analysis of the gtzan music genre dataset. In *Proceedings of the second international ACM workshop on Music information retrieval with user-centered and multimodal strategies*, pages 7–12. ACM, 2012.

[Sturm 2012b] Bob L Sturm. A survey of evaluation in music genre recognition. In *International Workshop on Adaptive Multimedia Retrieval*, 2012.

[Sturm 2013a] Bob L Sturm. The gtzan dataset: Its contents, its faults, their affects on evaluation, and its future use. *arXiv preprint arXiv:1306.1461*, 2013.

[Sturm 2013b] Bob L Sturm. On music genre classification via compressive sampling. In *Multimedia and Expo (ICME), 2013 IEEE International Conference on*, pages 1–6. IEEE, 2013.

[Sumner *et al.* 2005] Marc Sumner, Eibe Frank, and Mark Hall. Speeding up logistic model tree induction. In *9th European Conference on Principles and Practice of Knowledge Discovery in Databases*, pages 675–683. Springer, 2005.

[Swartz 2002] Aaron Swartz. Musicbrainz: A semantic web service. *Intelligent Systems, IEEE*, 17(1):76–77, 2002.

[Tekman and Hortacsu 2002] Hasan Gurkan Tekman and Nuran Hortacsu. Aspects of stylistic knowledge: What are different styles like and why do we listen to them? *Psychology of Music*, 30(1):28–47, 2002.

[Turnbull *et al.* 2007] Douglas Turnbull, Luke Barrington, David Torres, and Gert Lanckriet. Towards musical query-by-semantic-description using the cal500 data set. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 439–446. ACM, 2007.

[Tzanetakis and Cook 2002] George Tzanetakis and Perry Cook. Musical genre classification of audio signals. *Speech and Audio Processing, IEEE transactions on*, 10(5):293–302, 2002.

[Uitdenbogerd and Zobel 1999] Alexandra Uitdenbogerd and Justin Zobel. Melodic matching techniques for large music databases. In *Proceedings of the seventh ACM international conference on Multimedia (Part 1)*, pages 57–66. ACM, 1999.

[Unal *et al.* 2008] Erdem Unal, Elaine Chew, Panayiotis G Georgiou, and Shrikanth S Narayanan. Challenging uncertainty in query by humming systems: a fingerprinting approach. *Audio, Speech, and Language Processing, IEEE Transactions on*, 16(2):359–371, 2008.

[West and Cox 2004] Kristopher West and Stephen Cox. Features and classifiers for the automatic classification of musical audio signals. In *ISMIR*, 2004.

[White 1976] John David White. *The analysis of music*. Prentice-Hall Englewood Cliffs, NJ, 1976.

[Whitman and Smaragdis 2002] Brian Whitman and Paris Smaragdis. Combining musical and cultural features for intelligent style detection. In *ISMIR*. Citeseer, 2002.

[Wold *et al.* 1996] Erling Wold, Thom Blum, Douglas Keislar, and James Wheaten. Content-based classification, search, and retrieval of audio. *MultiMedia, IEEE*, 3(3):27–36, 1996.

[Xu *et al.* 2003] Changsheng Xu, MC Maddage, Xi Shao, Fang Cao, and Qi Tian. Musical genre classification using support vector machines. In *Acoustics, Speech, and Signal Processing, 2003. Proceedings.(ICASSP'03). 2003 IEEE International Conference on*, volume 5, pages V–429. IEEE, 2003.

[Xu *et al.* 2005a] Changsheng Xu, MC Maddage, and Xi Shao. Automatic music classification and summarization. *Speech and Audio Processing, IEEE Transactions on*, 13(3):441–450, 2005.

[Xu *et al.* 2005b] Min Xu, Ling-Yu Duan, Jianfei Cai, Liang-Tien Chia, Changsheng Xu, and Qi Tian. Hmm-based audio keyword generation. In *Advances in Multimedia Information Processing-PCM 2004*, pages 566–574. Springer, 2005.

[Yang *et al.* 2006] Kai-Chieh Yang, Clark C Guest, and Pankaj Das. Perceptual sharpness metric (psm) for compressed video. In *Multimedia and Expo, 2006 IEEE International Conference on*, pages 777–780. IEEE, 2006.

[Zhang and Kuo 2001] Tong Zhang and C-CJ Kuo. Audio content analysis for online audiovisual data segmentation and classification. *Speech and Audio Processing, IEEE Transactions on*, 9(4):441–457, 2001.

[Zheng *et al.* 2001] Fang Zheng, Guoliang Zhang, and Zhanjiang Song. Comparison of different implementations of mfcc. *Journal of Computer Science and Technology*, 16(6):582–589, 2001.

[Zhu and Shasha 2003] Yunyue Zhu and Dennis Shasha. Warping indexes with envelope transforms for query by humming. In *Proceedings of the 2003 ACM SIGMOD international conference on Management of data*, pages 181–192. ACM, 2003.

[Zwicker and Fastl 1990] E Zwicker and H Fastl. *Psychoacoustics: Facts and Models*, 1990.

[Zwicker *et al.* 1979] E Zwicker, E Terhardt, and E Paulus. Automatic speech recognition using psychoacoustic models. *The Journal of the Acoustical Society of America*, 65(2):487–498, 1979.

[Zwicker 1977] Eberhard Zwicker. Procedure for calculating loudness of temporally variable sounds. *The Journal of the Acoustical Society of America*, 62(3):675–682, 1977.