Classification of Ragas Using Psychoacoustic Features and Soft Computational Techniques

A Thesis

submitted in partial fulfillment of the requirements for the award of the degree of

Doctor of Philosophy

in

Department of Electronics and Communication Engg.

by

Chandanpreet Kaur
(Reg no: 951306005)

under the supervision of

Dr. Ravi Kumar
Associate Professor

THAPAR INSTITUTE
OF ENGINEERING & TECHNOLOGY
(Deemed to be University)

Thapar Institute of Engineering and Technology
Patiala, India
April 2019

Candidate Declaration

I hereby certify that the work, which is being presented in the thesis, entitled **Classification of Ragas Using Psychoacoustic Features and Soft Computational Techniques**, in partial fulfillment of the requirements for the award of the degree of **Doctor of Philosophy** in Electronics and Communication Engineering from Thapar Institute of Engineering and Technology, Patiala is an authentic record of my own work carried out during the period July 2013 to April 2019 under the supervision of Dr. Ravi Kumar. I have also cited the reference about the text(s)/figure(s)/table(s) from where they have been taken. The matter presented in this thesis has not been submitted elsewhere for the award of any other degree or diploma from any institution.

Date: 10 Feb 2020

Chandanpreet Kaur

Chandanpreet Kaur
Candidate

This is to certify that the above statement made by the candidate is correct to the best of my knowledge.

Date: 10/02/2020

Ravi

Dr. Ravi Kumar
Associate Professor
Supervisor

.........to my parents

Abstract

Classification of classical melodic structures by style, composer, genre, period, etc., is a rather complex task. The level of difficulty varies across melodic frameworks. It would be interesting to see how we can impart this ability to a machine. In this work, the problem of music classification is taken into consideration with special emphasis on the Raga classification. The challenges and obstacles in creating an automatic music classification system are acknowledged and studied. A new approach for clustering melodies in audio music collections of both western as well as Indian background and its application to genre classification. A simple yet effective new classification technique Mean Centered Clustering (MCC) is discussed. The proposed technique maximizes the distance between different clusters and reduces the spread of data in individual clusters. The use of MCC as a preprocessing technique for conventional classifiers like Artificial Neural Network (ANN) and Support Vector Machine (SVM) is also demonstrated. It is observed that the MCC based classifier outperforms the classifiers based on conventional techniques such as Principal Component Analysis (PCA) and Discrete Cosine Transform (DCT).

Subsequently, this dissertation reports an improved pattern matching technique for composer and raga classification using a fuzzy analytical hierarchy process-based approach. The technique makes use of class-specific patterns extracted from a pattern discovery technique known as Structure Induction Algorithm for r superdiagonals and compactness trawler. Further, to represent inexact matches a modified matching technique is proposed to assign weights to the exact matching scores in a probabilistic manner. Subsequently, the weighted scores are fuzzified to quantify the extent of match. Finally, the fuzzy scores are aggregated and classified on the basis of minimum Euclidean distance from an ideal solution in the pattern space.

Finally, the problem of classification of music structures by using different distributions is taken into consideration. Different popular probability distributions are taken into consideration for this task. The processing is done in both the time as well as frequency domain. MFCC coefficients are used as a basis to apply distribution estimation.

List of Publications

SCI Journal

1. Kaur, Chandanpreet, and Ravi Kumar, *"A fuzzy hierarchy-based pattern matching technique for melody classification"*, Soft Computing, pp. 1-18, 2018.

2. Kaur, Chandanpreet, and Ravi Kumar, *"Mean centred clustering: improving melody classification using time-and frequency-domain supervised clustering"*, Sadhana, vol. 44, no. 2, 2019.

International Conference

1. Kaur, Chandanpreet, and Ravi Kumar, *"Study and analysis of feature based automatic music genre classification using Gaussian mixture model"*, In 2017 International Conference on Inventive Computing and Informatics (ICICI), pp. 465-468. IEEE, 2017.

2. Kaur, Chandanpreet, and Ravi Kumar, *"Classification of melodic structures using fuzzified n-gram matching scores"*, In 2016 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), pp. 685-690. IEEE, 2016.

3. Kaur, Chandanpreet, and Ravi Kumar, *"Classification of Music Signal in to Genre using Probability Distribution Estimation"*, In 2018 International Journal of Electrical, Electronics and Data Communication, pp. 2320-2084, vol. 6, no. 4, pp. 36-39. IJEEDC, 2018.

Communicated

1. Kaur, Chandanpreet, and Ravi Kumar, *"n-Gram based Fuzzy Inference System for Classification of Melodic Structures"*, Sadhana, Springer

Acknowledgements

Firstly, I would like to express my sincere gratitude to my advisor Dr. Ravi Kumar for the continuous support of my Ph.D. study and related research, for his patience, motivation, and immense knowledge. His guidance helped me in all the time of research and writing of this thesis. I could not have imagined having a better advisor and mentor for my Ph.D. study. My sincere thanks also go to Dr. Neeraj Kumar who enlightening me on the first glance of research.

Besides my advisor, I would like to thank the rest of my thesis committee: Dr. Sanjay Sharma, Dr. Kulbir Singh, and Dr. Neeraj Kumar, for their insightful comments and encouragement. Their information has helped me complete this thesis. I would also like to thank Dr. Alpana Aggarwal, Head ECED, Thapar Institute of Engineering and Technology for providing useful support. I thank my fellow lab mates and hostel mates for the stimulating discussions. In particular, I am grateful to Dr. Gurnam Singh Professor and Head Gurmat Sangeet Chair and Dr. Amrinder Singh from Punjabi University, Patiala for helping me in sample collection. Their useful discussions made my work better.

Last but not least; I would like to thanks my family: my father and mother who encouraged me and prayed for me throughout the time of my research and my brother Sahib Singh for supporting me.

Chandanpreet Kaur

Table of Contents

List of Figures

List of Tables

List of Abbreviations

ADABOOST	Adaptive Boosting
AM	Amplitude Modulation
ANNs	Artificial neural networks
ARM	Auto-Regressive Modeling
ARS	Adaptive Round Semitones
ASE	Amplitude Spectrum Envelop
ATMM	Audio Technologies for Music and Media
DP	Dirichlet process
DSP	Digital Signal Processing
DWCH	Daubechies Wavelet Coefficient Histogram
ELM	Extreme Learning Machine
EM	expectation maximization
ETM	explicit time modeling
GHSOM	Growing Hierarchical SOM
GMM	Gaussian mixture model
HMM	Hidden Markov Models
HPCP	Harmonic Pitch Class Profile
ICMR	International Conference on Multimedia Retrieval
ISMIR	International Society of Music Information Retrieval
KNN	K-nearest neighbor
LDA	linear discriminant analysis
LPCC	Linear Predictive Cepstral Coefficient
LRR	lowest-rank representation
MFCCs	Mel-frequency cepstral coefficients
MIDI	Musical Instrument Digital Interface
MIR	Music Information Retrieval
MIREX	Music Information Retrieval Evaluation eXchange
MLP	multilayer perceptron
NB	naive Bayes
OSC	Octave based Spectral Contrast
PCP	Pitch Class Profile
QBSH	query by singing/humming
SC	Spectral Centroid
SCF	Spectral Crest Fcator

SF	Spectral Flux
SM	Statistical Moments
SOM	Self-Organizing Map
SPSF	Stereo Panning Spectrum Features
SR	Spectral Rolloff
SSC	subspace clustering
STFT	Short Time Fourier Transform
SVMs	Support vector machines
TDNN	time-delay neural network
ZCR	Zero-crossing rate

Chapter 1

Introduction

Music is what binds us together. Musical creations and performances are often complex and sophisticated and so is their processing and analysis. Recent innovations in information technology and especially digital signal processing have brought about tremendous changes in the way music content is used and accessed. Thus, computational musicology is a vibrant field aiming to surmount innumerable challenges [1]. Identification of individual melodic structures poses a plethora of challenges since there is inherent subjectivity in the way humans classify musical signals. Every musical phrase possesses some unique features which help humans distinguish one piece from another. These features hold in them the complete essence of both behavioral as well as signal-based properties of music [1, 2].

Music analysis is an emerging field of research in the signal processing community having a lot of exciting applications and challenging problems. Music is considered to be a sophisticated form of the audio signal which is precisely constructed using tones. Extracting the relevant information requires the application of different kinds of specialized methods involving music-specific characteristics e.g. harmony, rhythm, pitches, and instrumentation [1]. Music analysis opens avenues for a deep understanding of the composition and structure of music. This is essential to gain an insight into how humans perceive and enjoy music [3]. Classification of musical renditions into groups characterized through subjective human perception is an important step towards this. Music can be classified based on different parameters. Genre-based classification is one of the most popular one out of these and is extremely helpful in the task of Music Information Retrieval (MIR) [3]. Traditionally, expert listeners have done the job of classifying a musical rendition into genres and melodic frameworks. However, due to ever-increasing volume of the music data and advancements in signal processing machine based systems are employed for this task [2]. The classification based on the genre is a subjective and somewhat arbitrary one but a deeper insight into this process reveals that this process is based on types of instrumentation, texture and rhythmic structure of music. Similarly, renditions can be analyzed for identifying their belongingness to a particular framework. However, conducting research using unaltered versions of familiar melodies is complicated because the multilayered, interactive structure of music provides a wide range of musical features that could contribute to identification [4].

Raga in Indian classical music relates to the concept of melody. The term raga has a unique definition and this has little to do with the western concepts of tones and scales. A raga primarily prescribes the way a set of notes are to be inflected and ordered. In other words, raga is a melodic framework for improvisation during rendition considered in the Indian tradition to have the ability to affect the emotions of the audience [3]. A raga is characterized by a set of notes (at least five) and a set of ruler for the musician within which they are allowed to improve by recording and embellishing them. Indian musical tradition regards raga as a means to evoke certain emotions in an audience. The literal meaning of raga is colours, since its rendition is believed to colour the mind with a particular class of emotions. Longer the set of characteristic notes for a raga, more the scope of improvisation in it which in turn mean identification of the raga being played by a novice listener become more difficult.

Ganguli et. al. [5] proposed different methods for optimizing parameter settings for the subsequent string matching algorithm for Indian classical music. The task of raga music segmentation focusing on the preprocessing phase is discussed in [6]. In this study, the RapidMiner tool is used for the classification purpose and Jaudio is used for feature extraction in [6]. Authors in [7] depict a raga identification system using the Arohana-avarohana patterns. These patterns are identified by converting the given audio file into a sequence of notes. A novel data augmentation technique for Indian classical music based on the relative position of notes is presented in [8]. A convolutional neural network based approach is used for making a Raga classification system in [8].

This thesis is an effort towards the development of a robust and general purpose classifier for identification of raga in particular. Motivated by the need for an online music teacher for training novices in Indian music, this work exemplifies limited success in the improvement of pattern matching schemes for raga classification. In addition to it, the same techniques have been tested and validated a western music data for the purpose of composer classification. The results obtained underscore the universality of music as well as mathematics.

1.1 Concept of Music in different Cultures

The rapid advancement of internet technologies has made it easy for music listeners to access any type of music data online, including a huge amount of lyrics, music files, etc. In the 21st century, the music artists are promoted through different websites which are managed by their record companies, by themselves or by their fans. This makes essential to hold and tag music data. For the task of music classification, conferences like ACM International Conference on Multimedia Retrieval (ICMR), Audio Technologies for Music and Media (ATMM), International Society of Music Information Retrieval (ISMIR) and many more have worked on the advancement of music information retrieval (MIR). Although a large number of research projects and literature addressed the music information retrieval (MIR) over the past three decades, this topic is still highly popular among researchers [9–11].

However, most of the cultures around the world have their own way to classify music; how music is perceived varies from one culture to another. Each culture and race has its own history of music development over time. The history of Western music is primarily rooted in Greek and Roman antiquity, though music existed in virtually every culture long before this.

Western music refers to the traditions and melodic structures related to European culture. Therefore, music created in Europe, societies established by Europeans and the United States majorly contribute to western music. This results in very rich and diversified genres of music like jazz, country-western, hip-hop, blues, etc. [12]. Similar frameworks exist in Indian classical music. Music in Indian culture has existed for centuries. It is thought of as the first musical culture but the works on MIR in case of Indian classical music is rather limited.

Each genre in Indian classical music is considered to have its own well-defined structure and boundaries. Due to the enormous musical types and freedom of variations in them, the task of MIR and specifically, music classification is a challenging problem in this case. The difficulty is more amplified due to the fact that it is harder to differentiate if two singers are performing the same raga.

The raga is a remarkable and central feature of Indian classical music. Each raga is an array of melodic structures with musical motifs called the pakad or catch phase. This catch phrase is composed of at least five notes, and each raga has a distinct catch phrase. The musician can ofcause improvise while using the catch phrase in a melody based on particular raga. For instance, the motif or catch phrase for raga Yaman is [13]:

C D E F# G A B C

The definition of notes in Indian music requires the interpretations to be made based on the pattern of notes rather than individual notes [14]. The key difference between western and Indian musical system is that the western system is defined by twelve tones at equal intervals [15]. In Indian classical music, each class, traditionally known as raga, is based on pre-defined set of rules. Due to the large number of available variations in ragas and their types, automatic raga classification a tedious task. The reason behind this is that in Indian classical music, a lot of freedom is available with the artist. Due to this, large variations arise in performances of same music piece sung by different performers. Another key challenge is the fact that notes in Indian classical music are defined relatively. This motivated us to move away from directly trying to identify notes as features and conceptualize various features based on notes and their structural form. Typically the following features of musical signal are most commonly used in Music Information Retrieval.

1.2 Features of musical signals

Different features of musical signals are as follows [16]:

- Pitch: Pitch is a perceptual feature related to the frequency of a signal. The higher frequencies lead to the perception of a higher pitch [17].

- Timbre: It is the quality of the music signal that results in the differentiation of voices or musical instruments sounds from each other.

- Rhythm: Defines the repeating events in a musical piece. These events can be predicted by using energy, pitch and spectrum of data.

- Tempo induction: It measures the beats/minute and the interval between the beats.

- Energy: Energy defines the amount of signal present in a musical piece at a given time. It is calculated by taking a particular window of the signal (depending on the time of observation) and averaging the square of amplitude of the signal in that period. It is a very useful measure to distinguish between speech and music.

- Beat tracking: Beat is extracted from a musical piece using beat tracking. The beat histogram is generally used for this task.

- Fundamental frequency: It is defined as the minimum frequency at which the signal repeats itself. The periodicity of signal can be determined by using the fundamental frequency of signal.

- Spectral features: Fourier transform is commonly used to calculate the spectral features of a signal. a more efficient transform is Short Time Fourier Transform (STFT). It gives information about both frequency as well as time.

1.3 Music information retrieval (MIR)

Michael Kassler et.al [18] in the mid of 1960s gave the concept of MIR. Kassler and his co-authors were well ahead of the time in accessing the potential and promise of this field [18]. The researchers and societies have contributed towards summarizing, classifying, tagging/indexing musical piece and development of computational techniques. The work done by Bainbridge et.al. [19] in 1999 won the best paper award by Digital Libraries. The first high grants project was funded for MIR in 1999 to Wisman, Rusbridge and Griffin. The aim of this program was to ease the access to music collections [19]. In 2000, a project named OMRAS was granted to work in the fields of audio transcription, and document retrieval [20].

In 2000 the ISMIR organized their first conference. The conference included papers on: music classification using Competitive HMMs [21] and Mel Frequency Cepstral Coefficients for Music Modeling [22]; techniques for Automatic Music Transcription [23]; finding Motifs with Gaps; need and applications of Music Information Retrieval [24, 25]. MIR has emerged as one of the prominent research areas in music industry due to the advancements in signal processing techniques. Music pieces are annotated using several different meaningful text called tags. These tags can be used to identify the different classes of music like style, instrument used and genre etc. Therefore, music annotation is usually considered as a precursor to classification. The task of musical key detection in western music has similarities with raga identification in Indian music. A pitch profile based feature extraction method is utilized in [26] for key detection of musical data. In [27], a method for estimating diatonic scale and corresponding key from acoustic signals is explored.

Music classification refers to assigning a particular label to a given musical piece. This classification is generally based on the artist, year, genre, mood or cultural context. Music classification enables a user to search for the music of his choice without having a perfect knowledge about the technicalities of music [28]. This also results in effective management and the categorization of music data.

This task used to be predominately done by human experts in the past but recently, music classification by machines has become popular among research fraternity. To support and promote the task of MIR, Music Information Retrieval Evaluation eXchange (MIREX) is organized since 2004. This event contains competitions based on tasks in MIR. Most of these tasks in MIREX competitions are based on music classification [29]. Some of the tasks in MIRTEX based on music classification are as follows:

- Artist Identification [30]
- Genre Classification [13]
- Instrument Recognition [31]
- Mood Classification [29]
- Music Annotation [32]

1.4 Need of Music classification

Collection, management and retrieval of huge amount of music signals available online is becoming difficult each day. The information like artist, type, temp, genre etc. is required for this task [33].

The importance of automatic music classification can be enlisted as follows:

- Each music genres/class is composed/generated in a different way.
- Classification of music is helpful in MIR.
- Music classification has applications in different fields like archive management and entertainment.

1.5 Advantages of automatic music classification

Various advantages of automatic music classification are as follows [34, 35]:

- The task of music classification is done very effectively and in a fast manner using computers. On the other hand, a human expert requires the availability of real-time musical data.
- Another advantage of automatic classification systems is that they work on extracted features. These features can be extracted and stored efficiently by computers much faster than humans.
- Classification using computers is also more economical than manual classification by humans.
- Classification by humans requires more than one human experts while a single computer can classify thousands of songs.

1.6 Complementary approaches for analyzing music

Automatic music classification requires a contribution from fields like music theory, digital signal processing (DSP) and artificial intelligence [36].

The first and most important task in music classification is feature extraction. Based on the features extracted from a test signal (musical piece), a particular class can be assigned to it. Due to the increased use of digital media, the generic information in audio is usually represented by bits. This allows a perfect and direct reconstruction into an analog waveform. Another method to represent a melody is by using a high level model based representations like MIDI or MusicXML [37].

The conventional symbolic representation in the form of notes is seldom available. Instead, the digital audio file representation as a result of sampling the sound waveform is more popular. These samples cannot be directly used by the system for automatic classification because the amount of data present in these files is very large whereas the information contained in each sample is very small to be used independently for the task of music classification. Therefore, to make the system practical and reduce the processing required, features are extracted from the audio data. These features can be related to any property of music which is discussed in Section 1.2. After feature extraction, these features can be used further by a classifier. The most commonly used properties of music used in feature extraction are Timbre, Temporal features, Spectral shape features, Texture Window and High-level content-based features [31, 38, 39]. The most commonly used features used for timbre characterization are given in [34].

1.7 System for music classification

A workflow of the automatic music classification system is depicted in fig. 1.1. Given a particular music signal, the first step in an automatic music classification system is to collect ground truth data. This data includes audio recordings, symbolic recordings and cultural information related to a given signal. The next step involves the extraction of relevant features from the collected data. The choice of these features depends upon which class type one has to classify the musical data. These features are unique or distinguishable for every given class. Machine learning-based supervised or unsupervised learning can also be applied to the features for class labeling. Prior to that dimensionality reduction is also applied if required. Finally, the classifier is trained and a given musical piece is classified [40].

The work of Pachet and Cazaly [41] provide a deep analysis of various measure for quan-

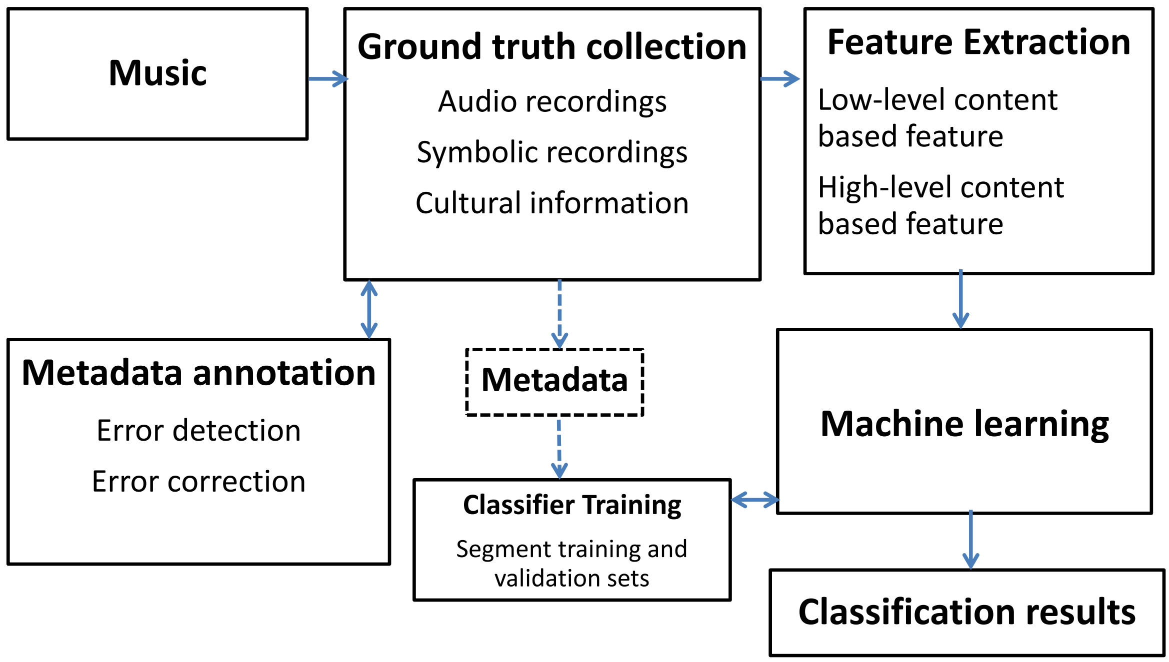

Figure 1.1: The steps performed for the task of automatic music classification.

tifying music similarity. The study provided an in-depth analysis of challenges involved in automatic music classification using an expert system. An expert system works on a set of rules that characterize a given class [42]. Even though expert systems are very efficient in the task of classification, yet with the increasing number of ruleset, their implementation becomes difficult. In the last decade; with the advancements in advance signal processing techniques, neural networks, fuzzy logic and pattern recognition algorithms; expert systems have emerged as the promising candidate for automatic music classification [43–45]. The expert systems can be implemented using either supervised or unsupervised approaches depending on if preliminary information about the classification task is available or not [46]. Similarity measures used in these approaches include calculating the distance between two feature vectors. Most commonly used distances are Euclidean distance, Kullback-Leibler divergence or relative entropy. In order to calculate these similarity measures, several well known models like Gaussian and Gaussian mixtures (GMMs) [19], asymptotic likelihood approximation [47, 48], hidden Markov models (HMMs) [49] are utilized.

Different types of clustering algorithms are also a part of expert systems. One of the most popular and simplest clustering algorithm is the K-means clustering algorithm. Agglomerative hierarchical clustering is another popular clustering algorithm [49]. An optimized k-means clustering technique using bat algorithm was given in [50]. A framework for EEG and MEG source localization was given in [51]. The self-organizing map (SOM) is unsupervised artificial neural network used in clustering algorithms. The growing hierarchical SOM (GHSOM) is an advancement in SOM which works on multiple layers. The steady-state security analysis was performed in [52] using ANN. Several notable works on supervised learning approach are [13, 53–56].

For large datasets, deep neural network (DNN) has been used as a powerful tool. The effects of noisy labels on the performance of music tagging system using DNN is discussed in [57]. The use of DNN in automatic music recommendation has been studied in [58]. The results are compared with the traditional bag-of-words representation approach. In [59], an automatic feature extraction system based on Deep Belief Network (DBN) and Discrete Fourier Transforms (DFTs) have been introduced with which the problem of genre recognition has been solved. It was observed that the learned features perform significantly better than MFCCs. A music genre classification system based on DNN is discussed in [11] in which the potential dependency on the tuning frequency has been discussed. The classification accuracy was studied based on the pitch shifting of the audio data. The linear discriminant analysis (LDA) is one of the popular supervised learning technique. LDA with adaptive boosting (Adaboost) has been used in [60]. In [60], a Fishers criterion multiclass LDA is used to reduce the dimensionality of the classification problem before modeling with a Gaussian distribution. Support vector machines (SVMs) are another type of supervised learning algorithms [61]. In [61], SVMs are used for genre classification with a Kullback Leiber divergence.

1.8 Applications of automatic music classification

Some of the notable applications of automatic music classification are as follows [62]:

- To create personal collections
- Generating playlist according to user specifications
- Audio fingerprinting, monitoring and automatic labeling in radio-stations
- Music Recommendation
- Multiple relationships in music stores and library databases

The most basic approach of audio classification is the separation of the audio signal into the parts where speech, music or environmental sounds exist [63]. What the latest MIR work tries to add is the separation of music itself, into several other classes. The classification can be applied, based on any of the dimensions of the items such as their orchestration. However, the most usual classification task is the genre and melody classification. Pieces are separated into the various genres and sub-genres, which are the way that most users tend to, understand and classify music. The size of the audio data that are necessary to have good classification results is the main concern, regarding this task.

1.9 Literature survey

This section describes the work done by various researchers so far in the field of music analysis. The goal of music analysis is generally to characterize a given musical signal based on the motive and theme of music. Changsheng et al. [64] presented an approach for music structure analysis based on note onset and time tempo. Yuting Qi et al. [65] have developed a discrete HMM model in a Bayesian setting using Dirichlet process (DP) priors, which has the advantage of avoiding the need to select the number of mixture components, through the encouragement of parameter sharing. Meinard et al. [66] provides an overview of some signal analysis techniques that specifically address musical dimensions such as melody, harmony, rhythm, and timbre. Lu Ren et al. [67] proposed a Bayesian dynamic mixture model for music analysis. Tsukasa et al. [68] proposed a method in which they consider the music genre as a base for music analysis. SVM based classifiers were used in this task.

The music style is one of the features that people used to classify music. Lippens et al. [69] worked on the problem of automatic musical genre classification. The results obtained in [69] were compared with results from human experts. Changsheng Xu et al. [70] gave an SVM based automatic music classification system. In this task, they separated pure music and vocal music using training data. Zhouyu Fu et al. [71] discussed the use of naive Bayes (NB) classifiers for music classification and retrieval. They proposed two NB classifiers, namely Naive Bayes Nearest Neighbor (NBNN) and NB Support Vector Machine (NBSVM). Jing Lu et al. [72] proposed a method for music classification based on multi-class SVM and MFCC. Simone et al. [73] worked on the use of Extreme Learning Machine (ELM) in problems related to music classification. In [74], a software system for automatic classification of MIDI files is discussed. This hybrid classification is based on hierarchically organized taxonomies of musical genres that use hierarchical, flat and round-robin classification. McKay and Fujinaga [75] developed a library of 160 high-level features for automatic music classification or to evaluate musical similarity. A software package named jSymbolic is also proposed in this paper.

Ulas and Engin [76] used the dynamic timbral texture features for musical genre classification. Two new classifiers are utilized to improve genre classification. Chathuranga and Jayaratne [77] proposed an approach using the SVM classifier. Authors in [39] proposed an automatic music genre classification system by using three different feature sets for representing timbral texture, rhythmic content and pitch content. A classification of 61% for ten musical genres is reported in [39]. Li and Ogihara [15] worked on hierarchical taxonomy music classification. Aryafar et al. [78] used short time representations of audio signals for genre classification using SVM classifier.

Kuo and Shan [79] presented a music filtering system by learning the users preference on melody style. Wang et al. [80] proposed an adaptive round semitones algorithm to convert pitch contour into music notes. Ryynanen et al. [81] propose an automatic melody transcription system. This system was used in karaoke application. Through this method, user singing can be converted into the original melody. Pollastri and Simoncelli [82] use HMMs for detecting and recognizing the style of a particular composer. Downie in his work in 2003 [83] explained the use and importance of MIR in the music industry. Heittola [84] worked on the problem of genre classification for large datasets. Futrelle and Downie in [85] summarized the work done in the last few decades on MIR techniques of music. Fabbri [22] stated following rules for music genre classification:

- Formal and technical: Content based practices
- Semiotics: Abstract concepts that are communicated
- Behavior: How composers and performers behave
- Social and Ideological: Links between genres and demographics (age, race...)
- Economical: Economic systems that support specific genres.

Cook [67] described the features which are used by humans to describe music. In [68], features used by humans for music genre classification are described [68].

String matching techniques for musical genre classification have been used in the past by various researchers [86, 87]. String methods like edit distance, normalized compression distance (NCD) and string subsequence kernel method (SSK) were used in [88] for the classification of folk tunes into genres of the dance types of the tunes. The task of polyphonic music classification was studied in [89] and [90]. A global feature model with standard machine learning classifiers is compared with a monophonic n-Gram model for the task of composer recognition. A dataset of 207 Haydn and Mozart strings were used to form the database. An extension of the multiple viewpoint method for music prediction and generation was proposed in [91] to predict the genres of unlabelled Basque folk tunes. The algorithm was evaluated on four folk tune datasets for the task of genre and region classification. n-Gram matching is a standard string matching technique widely applied in the field of computational linguistics and forms the basis of ubiquitous plagiarism detection software [92–94] and structured prediction [95].

Methods other than string matching are being reported for genre classification. Lee et al. in 2009 [96] reported an information fusion approach using feature level fusion and decision level combination to improve the genre classification accuracy. Rao et al. in 2014 used Dynamic time warping and Hidden Markov Model (HMM) for the classification of raga music [97]. However, when we talk about the existence of a standard template for a class of music, there is enormous subjectivity involved interpreting such melodic structures. In other words, the template matching scores are rather not infallible indicators of a particular class. The degree of belongingness of a piece of rendition to a particular class is open to the interpretation of a trained listener.

It is thus obvious that soft decision models are a better candidate for obtaining the classification of such structures [97]. Several rule-based systems are often reported in the literature to solve complex classification problems where crisp modeling of the system is not essentially productive [98]. These systems can be categorized into expert systems and fuzzy systems. Fuzzy logic based classifiers have been extensively used in the literature due to their high performance and robustness in dealing with real-world complex classification problems [98–101]. Classifiers based on fuzzy logic utilize linguistic variables and fuzzy sets to create a complex nonlinear relationship between input and output variables. An efficient fuzzy classifier with feature selection based on fuzzy entropy for pattern classification is given by Lee et al. [98]. The performance of a hybrid fuzzy genetic machine learning method is examined for multidimensional pattern classification by Ishibuchi et al. [102].

The problems of pattern induction and matching for melodies were discussed by Klapuri in [103]. Liu and Liao [104] provided a brief study on bibliometric analysis of fuzzy decision research. This study was focused on finding underlying patterns, hesitant fuzzy set and fuzzy environment. Pollastri and Simoncelli [105] present a model for identifying the composer using a Hidden Markov Model (HMM) based classifier. The authors have extracted relevant themes from musical pieces of each composer using rhythmic contour based motif extraction. Motif based clustering is used in [105] and [106] for music classification tasks. Dynamic Time Warping (DTW) has been used in [107] for similarity matching in order to tolerate more variability. Apart from these techniques, n-Gram matching has also been extensively used for speech processing [108–111].

In [111] authors studied the importance of using higher order variable length n-Gram models and addressed the issues related with it. Bimbot *et al.* [112] proposed a variable length sequence model as multigram model and compared it with conventional n-Gram model. Siu and Ostendorf [113] investigated a variable n-Gram algorithm and extending the algorithm to handle conversational speech characteristics like repetitions and other disfluencies. Chen *et al.* [114] report an experimental study on variable length n-Gram model and demonstrate that their model extracts essential information of a sequential database.

There are several studies describing the use of weighted average of variable length phrase matchings against the reference translations [115, 116]. The n-Gram model requires a predefined pattern which is matched with the test string. In order to calculate this pattern, several pattern discovery algorithms are used by researchers. A pattern induction algorithm was proposed by Crow and Smith [117] which computed all the maximal repeated factors in a string. Later on, Cambouropoulos [118] introduced Sequential Pattern Induction Algorithm (SPIA). SPIA was more efficient algorithm based on a partitioning technique. Hsu *et al.* [119] gave an algorithm to find repeating factors in strings representing monophonic melodies using a dynamic programming technique. Rolland *et al.* used an extended toolbox of editing operations for discovering repetitions [120].

Several algorithms like Structure Induction Algorithm (SIA) [121], SIARCT [122], Structure Induction Algorithm for r superdiagonals and Compactness Trawler categorization fingerprinting (SIARCT-CFP) [123] etc have been developed to discover the repeated motifs, themes and sections in music. SIA is basically a geometric approach to pattern discovery which involves converting each note to a point in pitch-time space. This approach has been found to be quite effective in discovering repeated patterns inside a piece of music especially polyphonic music. Subsequent evolution of SIA into SIARCT-CFP has helped overcome many limitations of the original SIA one of them being a high proportion of false positives. Motivated by the efficacy of SIARCT, we have used this algorithm in this work for extracting intra opus relevant motifs. Coming back to the pattern matching part, fuzzy logic has been incorporated to improve n-Gram based techniques [124, 125] though not for musical analysis. Alzahrani *et al.* [125] gave taxonomy of plagiarism detection methods including: character n-Gram based, vector-based, fuzzy based structural based, semantic based and cross-lingual based techniques. Chen and Rosenfeld [126] gave the comparison of maximum entropy smoothing technique with conventional techniques for smoothing n-Gram language model. They demonstrated that the fuzzy maximum entropy method used to smooth maximum entropy gives better performance. Lerch in

[17] has explored the topic of audio content analysis in detail. Different features such as statistical properties, spectral shape, technical/signal properties, and intensity properties are also studied in [17].

However, most of the works employing fuzzy models have been confined to similarity matching in plagiarism detection software. It is thus natural to envisage extension of fuzzy logic based models to classification of melodic structures. In this regard multiple criteria decision models (MCDM) making use of fuzzy logic require special mention. MCDMs have emerged as robust paradigms which evaluate alternatives with respect to a multitude of criteria. To deal with fuzziness and uncertainty in multiple criteria decision making the fuzzy analytic hierarchy process (AHP) has found remarkable applications [127, 128]. Ren et al. [129] present a new intuitionistic multiplicative group AHP for managing complex multicriteria group decision making problems with individual intuitionistic multiplicative preference relations. The issues of uncertainty and rank reversal paradox in MCDM are studied in [130]. AHP basically works on a hierarchical model which derives the weights for each criteria from pairwise comparison matrices. The authors have recently employed a fuzzy AHP model for identification of a small set of Indian Ragas and obtained encouraging results [131]. The present work seeks to substantially improve the model reported in [131] to make it more general purpose and analytically justified.

The music classification systems based on content are utilized to cover the gaps by taking labels from low level features. Table 1.1 shows the summary of low-level features which are used by various researchers in music classification.

The K-nearest neighbor SVM, and GMM classifier are the two most popular classifiers used in music analysis. Apart from these, several other classifiers are also used such as logistic regression [143], linear discriminant analysis (LDA), nearest centroid (NC), artificial neural networks (ANN) [85] and sparse representation based classifier. Table 1.2 gives a comprehensive survey of different features used in music classification.

Table 1.1: Summary of common low-level features used in music classification

Feature Type	Features-Level	Class	Used in
ZCR	Low-Level Feature	Timber	[2, 132, 132, 133]
MFCC			[132–135]
Spectral Centroid (SC)			[2, 132, 132, 133, 136]
Spectral Flux (SF)			[2, 132, 136]
Spectral Rolloff (SR)			[2, 132, 132, 133, 136]
Spectral Bandwidth (SB)			[132, 133, 136]
LPCC			[135]
DWCH			[2, 137]
SFM			[138–140]
ASE			[135]
FCC			[133]
SCF			[138–140]
SPSF			[141]
OSC			[136]
ARM		Temporal	[134]
SM			[2, 133]
Amplitude Modulation (AM)			[134]
PCP	Mid-Level Feature	Pitch	[132]
Spectral modelling synthesis (SMS)			[138, 142]
HPCP			[36, 40, 45]

Table 1.2: Summary of features used in music classification

Reference	Features	Algorithm	Classification
Dannenberg et al., 1997 [144]	MIDI key number	Naive Bayes	90
Cruz-Alcazar 1998 [145]	melodic intervals + duration	Grammatical Inference	95
Chai 2001 [146]	pitch contours	HMM	59
Shan 2002 [79]	chord degrees progression	Associative rules	84.2
Basili et al., 2004 [147]	changes in meter/time	Naive Bayes	41
Cruz-Alcazar et al., 2003 [148]	melodic intervals + duration	Grammatical Inference	92.3
Li, 2004 [149]	statistics from pitch, duration	SVM + hierarchical	75.7
Changsheng et al., 2005 [70]	LPCC	SVM method, traditional	good
Bergstra, 2006 [133]	STFT, FFT, MFCC, LPC	AdaBoost.DT	82
Lee et al., 2009 [31]	MFCC,ASE,OSCFP,LDA	NC	91
Panagakis, 2010 [150]	CR, NTF	SVM	78
Fu et al., 2010 [151]	MFCC,ASE,OSC,Beat, chord	SVM (MKL)(SG)	91
Emiru et al., 2011 [9]	Timber, Rhtytm, Bass-line	k-means	76
Simone et al., 2013 [73]	MFCC	Neural Net, ELM	76
Huang et al., 2014 [152]	Intensity, pitch, timbre etc.	SVM	97
Jia-Min, 2015 [153]	SSD, MFCC, OSC	SVM	57
Loris et al., 2015 [154]	SVM and ADABOOST	SVM	85
Yandre et al., 2017 [155]	Binary Patterns	CNN, SVM classifiers	92
Farrokhmanesh, 2018 [156]	MFCC, Chromogram	KNN	95

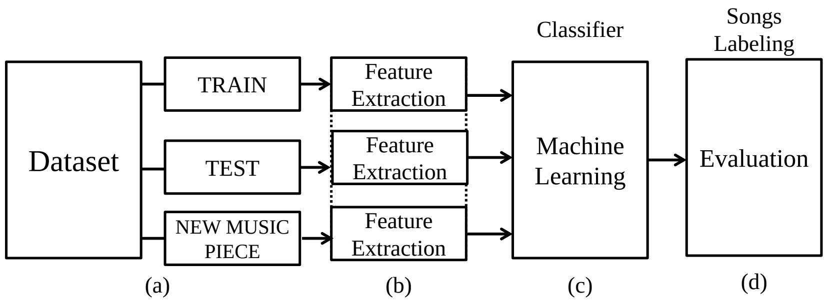

Figure 1.2: Block diagram of proposed framework: (a) Dataset collection, (b) Feature extraction, (c) Machine learning algorithm (d) Evaluation.

1.10 Gaps in study

Based on the literature review several gaps in the present state of the art were identified. Some major limitation of popular classifiers and pre-processing can be stated as follow:

- In HMM it is not possible to predict the prior probability of seeing an arbitrary observation. And this claim is only valid if the probability of a residue is independent of the probabilities of its adjacent state.
- Using the MFCC feature while building a vocal region model STFT is used to analyses the music domain coefficients which provide a single resolution to the entire signal which in turn results in the loss of time resolution.
- Although SVM is an efficient analytical technique for classification. However, it works better as a binary classifier. To perform a multi-class classification, one against many classification scheme can be used. This method is computationally expensive and runs slow.
- In K-mean clustering it is difficult to guess the initial value of K. It also does not work well with non-globular clusters.

- Back propagation trained ANN being very popular for classification purpose, it was quite surprising to see a lesser number of papers on music classification. This has served as a motivation for the author to try investigating the efficacy of ANN's apart from other soft computational techniques.

In a nutshell, the author identified a pertinent research gap in terms of the lack of a general-purpose classifier which could extract relevant motif from musical pieces across genre, composition and melodic frameworks (ragas).

1.11 Objectives and Methodology

Traditional data processing technique uses statistical information for specialized processing music signals it is natural that we would like to have distinct musical features at our disposal. However, many of the prominent music features not being well defined, it is desirable that we make use of statistical features alone for identifying generic signatures. In light of the research gap identified and the above discussion the following objectives can be defined for our problem.

- Identification of statistically relevant features from a wide variety of raga signals.
- Development of novel clustering techniques based on the above features.
- Estimation of the underlying probability distribution of raga signals in time and frequency domain.
- Construction of an appropriate mixture model for obtaining unsupervised genre classification and its comparison with ANN model [157].

As the first step, representative pieces of music were collected from online repositories as reported in section 1.13 of the thesis. Both western and raga data were obtained in the form of MIDI files. The selection of music files was completely random to rule out any confirmation bias. However, in the case of raga data, the help of a human expert was taken to confirm the purity of the raga rendition (a condition which is ought to be fulfilled in Indian musical tradition). During the course of research basically, three types of classification strategies were adopted by the author. The first approach consisted of training an ANN-based classifier with pre-processed features of raw audio signals extracted from MIDI files. In the course of this work, the author has found a supervised clustering technique to work particularly well with ANN classifiers. The second strategy was pattern matching which required extraction of a representative feature from each raga/genre/composer.

For this, the author made use of already reported Structure Induction Algorithm to find a representative motif for every MIDI set of notes fed to it. The real contribution of the author lies in developing a weighted pattern matching technique which was further improved upon fuzzification. In order to attempt unsupervised genre classification, the author estimates the probability distribution of music belonging to different genres in time and frequency domain. This work makes use of MATLAB 2017 for simulation. In particular, the ANN Toolbox, Fuzzy logic tool-box and signal processing toolbox of MATLAB were used.

1.12 Novel Aspects

A lot of work has been done in literature in this regard but the efficiency of the presented algorithms is not considerably good especially for unsupervised algorithms. Secondly, most of these algorithms focus on only melodies from the same background like classical western music, Chinese music, Indian music etc. So the development of an algorithm that can handle melodies from different backgrounds is also of prime interest. An emphasis has been put on the flawless classification of Indian ragas; however, the present work also addresses composer by composer classification of western melodies.

The contributions made by this work are enlisted as follow:

- Several classifiers and preprocessing techniques have been tried out to clarify pieces of musical rendition (vocal) in terms of genre, composer and ragas.
- We report a novel pattern matching based classifier that can label music across genre (chapter 4).
- Both time and frequency domain features of music have been needed for obtaining a classification.
- Soft decision models (specially fuzzified one) have been successfully implemented for raga and composer classification.

The body of work reported in this thesis and the results obtained could serve as an exemplar for more specific future work in this area of music classification.

1.13 Proposed Framework

Fig. 1.2 shows a block diagram of the proposed framework used in this study. The block diagram consists of four major blocks. The first block consists of dataset creation and management. The dataset is divided into training and testing datasets. The second block is the feature extraction block which calculates related features that are used for further processing. Thereafter, in the classifier block, machine learning is used. Further, based on the results provided by machine learning, labeling of a song is achieved in the evaluation block.

1.14 Data Acquisition

The experiments have been performed on different datasets collected from a wide spectrum of published musical work. Chapter 2 on Mean Centered Clustering (MCC) uses audio as an input format. Two datasets are used for the purpose of validating the results of MCC given in Chapter 2. These datasets are the Raga dataset containing audio files of classical Indian music based on Raga class and MIDI dataset containing MIDI ISMIR database music audio benchmark data set containing western music audio files.

The first data set named Raga dataset consists of 50 songs from 5 classes from different Ragas Asa, Basant, Malhar, Shiri and Gauri which belong to classical Indian music. Ten songs from each class were taken into consideration. Three clippings are extracted from each song. The raga samples have been obtained from www.searchgurbani.com whose excerpts are taken from Guru Granth Sahib: An Advanced Study [158], a book considered a scholarly source of authentic information on Indian religious music and poetry.

The second dataset called as MIDI dataset contains MIDI ISMIR database music audio benchmark data set [81]. This dataset contains fifty western music audio songs of twenty-second each, ten from each genre pop, folk, country, electronic and blues. This is done in order to validate the performance of the proposed technique on musical pieces other than ragas.

Chapter 3 and 4 related to n-Gram techniques use symbolic MIDI input data formats. Notes from these MIDI files have been used for the task of n-Gram based classification. Along with this, the onset time in tatums and the chromatic pitch corresponding to MIDI notes are also used to extract the catch phrase. Two datasets have been taken for this purpose. The first dataset called Dataset-1 contains melodies from different composers related to western music. Melodies created as a result of the skillful work of a specific composer hold the essence of that composer in themselves. Hence, it is assumed here in this work that renditions by one composer belong to one class. Melodies from Saarland Music Data (SMD) [73] are used to construct Dataset-1.

The second dataset for this task is called Dataset-2 contains melodies from Indian clas-

Table 1.3: Dataset-1 containing classes belonging to famous composers of western classical music.

Composer Name	Class
Achille-Claude Debussy	W1
Franz Peter Schubert	W2
Joseph Haydn	W3
Johannes Brahms	W4
Franz Liszt	W5
Wolfgang Amadeus Mozart	W6
Maurice Ravel	W7
Camille Saint-Saens	W8
Sergey Vasilyevich Rachmaninov	W9
Frederic Chopin	W10

sical music which is based on ragas. The melodies belonging to a particular raga are assumed to be a unique class here. The data samples of melodies are taken from http://www.anunaadacademy.com/ (Anunaad Music Academy, 2015) for Dataset-2 which hosts a repository of Raga based music files [159].

These datasets consist of notes from 100 different songs; 5 from each class of ten different composers (Dataset- 1) and Ragas (Dataset-2). The note sequences have been verified by matching them with the annotations by human experts. A window of 50 notes from each song is extracted. All the results shown in this work have been validated by subjecting the data to fivefold cross-validation scheme. The extraction of a 50 note window for the test string has been done five times from the entire length of the song. Hence, a dictionary of a total of 500 strings of notes was used. We have designated a set of 500 test strings in five folds as DW 1, DW 2, DW 3, DW 4, and DW 5 for Dataset-1 and DR1, DR2, DR3, DR4, and DR5 for Dataset-2 respectively. The classes of Dataset-1 were annotated with $W1, W2, \cdots, W10$ and of Dataset-2 were annotated with $R1, R2, \cdots, R10$ respectively for distinguishing results of both datasets. The details of both datasets are given in Table 3 for western and Table 4 for Indian classical music.

The details of both datasets are given in Table 1.3 for western and Table 1.4 for Indian classical music. Apart from these two datasets, different other datasets were also used whose explanation is as follows.

In Chapter 5, audio signals are used as input. The results in Chapter 5 are calculated

Table 1.4: Dataset-2 containing classes belonging to Raga from classical Indian music.

Raga Name	Class
Bhairavi	R1
Basant	R2
Malhar	R3
Yaman	R4
Todi	R5
Peelu	R6
Bahar	R7
Gaud	R8
Behag	R9
Asa	R10

on four different standard datasets named ballroom dataset [160], SLAC dataset [161], Codaich dataset [162], and Bodhidharma dataset [163]. The ten genres in these datasets are Blues, Classical, Country, Disco, Hiphop, Jazz, Metal, Pop, Reggae, and Rock. The excerpts of the dataset were taken from [69]. From each music track, the sound signals over a period of 30 seconds after the initial 30 seconds were extracted in MP3. All the melodies in the dataset are converted to 22050Hz, 16-bit, mono audio files. The complete dataset was partitioned randomly into subsets, i.e. training set consist of roughly 80% samples, and 20% for validation and test sets each. Further fivefold cross-validation is employed in these datasets to remove the problem of overfitting.

1.15 Organization of the thesis

In the subsequent chapter, we describe a simple preprocessing technique for clustering of melodies into their generic classes. The technique described in chapter 2 as MCC was found worth well with low complexity. However, it required a large number of labeled samples and a subsequent classifier whose performance affected the overall classification results.

Chapter 3 gives a brief study of the n-Gram technique for pattern-based algorithm for effective melody classification. The extraction of catch phrase with Structure Induction Algorithm (SIA) based algorithm is given in the chapter for pattern-based matching classification.

In Chapter 4 the traditional n-Gram algorithm is improved by developing a new n-Gram based approach named weighted n-Gram in this thesis. The weighted n-Gram, when used with traditional n-Gram, is used for melodic classification. This algorithm works well for both classical Indian music as well as classical western music which are the two very popular music forms. The requirement of predefined catch phrase by the n-Gram algorithm is also removed by using the previously reported (SIA) based algorithm.

Thereafter, the weighted n-Gram and traditional n-Gram are clubbed together and the input to is different classifiers that result in a system that can classify a given melody file from both classical Indian as well as classical western music background into their generic classes. Extensive simulation results using MATLAB are provided to verify the efficiency of the presented algorithms. Chapter 5 reports unsupervised classification strategy using Gaussian Mixture Model (GMM) for identification of musical genres. This chapter is appended in the form of a record of a GMM classifier trained with different temporal and spectral features of individual musical genre. It should be mentioned here that GMM models seem to perform rather poorly for Raga classification task. Secondly, the last section of chapter 5 reports probability Density Estimation of musical signal from several genre in order to give future work as a background. Chapter 6 gives the conclusion and future work of the dissertation.

Chapter 2

Mean Centered Clustering

In this chapter, we introduce a new approach for the classification of audio music collections of both western as well as Indian music and its application to both raga and genre classification. A simple yet effective new classification technique Mean Centered Clustering (MCC) is discussed. The proposed technique maximizes the distance between different clusters and reduces the spread of data in individual clusters. The use of MCC as a preprocessing technique for conventional classifiers like Artificial Neural Network (ANN) and Support Vector Machine (SVM) is also demonstrated. It is observed that the MCC based classifier outperforms the classifiers based on conventional techniques such as Principal Component Analysis (PCA) and Discrete Cosine Transform (DCT). Extensive simulation results obtained on different datasets of western genre (ISMIR) and classical Indian ragas are used to validate the efficiency of the proposed MCC based clustering algorithm and ANN/SVM classifiers based on MCC. As an additional endeavor, the performance of MCC on preprocessed data from PCA and DCT is studied. Based on simulation results, it is concluded that the application of MCC on DCT coefficients resulted in the highest overall classification success rate over different architectures of the classifiers.

2.1 Motivation

Classification of melodic structures poses a plethora of computational challenges due to poor separability of musical data in the pattern space. It is a well known fact that the performance of any classifier depends greatly on the type of data given as input. Therefore, it is quintessential to apply preprocessing to the data before applying classification. This results in the linear separability of the data in pattern space. This has served as a motivation for the author to apply centering and scaling method on raw data generated from the musical signal in time and frequency domains. The data preprocessed this way was subject to processed data that the overall classification percentage improves when MCC is applied to raw data. But in some cases, the classification percentage is low. To improve this performance, MCC is applied to DCT processed data. For example, in the case of ANN classifier, when the number of hidden layers is increased, the classification percentage improves for the case when MCC is applied on DCT processed data. Experiments were done on other combinations like PCA on MCC but the results were rejected due to poor classification performance of such combinations. Conventional techniques like PCA and DCT lack in creating linear separability which introduces improper and overlapped decision boundaries.

2.2 Related work

There has been many work reported in the literature for music classification by using different clustering techniques. There are many variants of clustering algorithms, which fall into one of two groups: hierarchical [164] and non-hierarchical [165]. The resulting vectors corresponding to the various files are then classified or clustered using existing classification software, based on various standard statistical pattern recognition classifiers, k-prototypes algorithm [166], Bayesian classifiers [167], hidden Markov models [168], ensembles of nearest neighbor classifiers, or neural networks. Purohit et. al. [169] introduced a new efficient approach towards the K-means clustering algorithm. They proposed a new method for generating the cluster center by reducing the mean square error of the final cluster without a large increment in the execution time. It reduced the means square error without sacrificing the execution time.

Authors in [170] proposed brain tumor segmentation using K-means clustering and fuzzy c-means algorithm. Yedla et. al. [171] proposed Enhancing K-means clustering algorithm with the improved initial center. A new method for finding the initial centroid is introduced and it provides an effective way of assigning the data points to suitable clusters with reduced time complexity. They proved their proposed algorithm has more accuracy with less computational time comparatively original k-means clustering algorithm. Authors in [172] have indicated the ability of clustering to group documents with respect to topic relevance; such findings are the basis for the clustering hypothesis. To group a set of documents into clusters of documents relevant to dierent instances of a topic requires clustering with respect to instance relevance.

Authors in [173] use Self Organized Map (SOM) technique and preprocessed the data of raw music raga files from that they utilized the frequency domain sampling by FFT which shows highly co-related with the clusters they predicted to belong to. In [76], Kirthika and Chattamvelli proposed a general system architecture for clustering and classification purpose of ragas.

In [174], a method of music segmentation is proposed which is based on the hierarchical labeling of spectral features. A method based on strong changes of timbre to indicate possible section boundaries of music was given in [175] and the method proposed in [176] is based on calculating the reoccurrence of sections of a particular type for clustering melodies.

2.3 Novel Aspects

The novel contribution of this chapter is the introduction of a clustering technique. This technique centers the training samples of the musical piece about its cluster mean called Mean Center Clustering (MCC) turns improving the class separability of the samples in pattern space. This clustering technique has been applied both on the time domain and frequency domain samples. The idea behind the proposed technique is derived from the context of image de-blurring in which it is observed that a clustered pattern space improves the performance of the neural network. A clustered pattern space contains different distribution characteristics for different classes of data which makes the neural network search for common ground while approximating the underlying function. A network trained this way is less likely to suffer from overfitting and more likely to yield good performance with unlabeled samples [177].

Finally, the use of MCC as a preprocessing technique for conventional classifiers like Artificial Neural Network (ANN) and Support Vector Machine (SVM) is also demonstrated. It is observed that the MCC based classifier outperforms the classifiers based on conventional techniques such as Principal Component Analysis (PCA) and Discrete Cosine Transform (DCT). The efficiency of the proposed clustering technique is validated using extensive simulation results.

2.4 Challenges in melody clustering and classification

Clustering and classification of melodic structures pose a plethora of computational challenges due to poor separability of musical data in the pattern space. It often becomes necessary to make the data linearly separable in the pattern space before subjecting it to a classifier.

- Very high dimensionality of the data samples in a single melody file
- Very large size of the databases
- Overlapping boundaries of cluster classes

It has also been observed from the results provided in the next section that conventional techniques like PCA and DCT lack in creating linear separability which introduces improper and overlapped decision boundaries. This has served as a motivation for the authors to envisage a technique that could improve the class separability of the data in the pattern space.

2.4.1 Principal Component Analysis (PCA)

PCA is a standard statistical technique that can be used to reduce the dimensionality of a data set of different classifiers on different musical feature sets to determine the genre of a given music piece, without much loss of information. PCA is an unsupervised method, which makes no use of information embodied within the class variable [178]. A data matrix (D) is constructed by taking songs of different classes as column vectors. Then, the covariance matrix R for matrix D is calculated as [179]

$$R = E[DD^T] \tag{2.1}$$

Further, (2.1) can be written in terms of eigen values $\lambda_1 > \lambda_2 > \cdots \lambda_j > \cdots \lambda_m$ and the associated eigen vectors $q_1, q_2, \cdots q_j \cdots q_m$ as

$$Rq_j = \lambda_j q_j; \qquad j = 1, 2, \cdots, m \tag{2.2}$$

Equation (2.2) can be re-written as

$$RQ = Q\Lambda \tag{2.3}$$

Where $\Lambda =$ Diagonal $[\lambda_1, \lambda_2, \cdots, \lambda_j, \cdots, \lambda_m]$ and $Q = [q_1, q_2, \cdots q_j \cdots q_m]$. The projection of the sample value x of the random vector X onto the respective eigen vector is given as

$$a_j = q_j^T x = x^T q_j \tag{2.4}$$

Where a_j are called principal components. PCA seems to be an essential pre-processing stage since our data is multidimensional. Though the application of PCA mitigates the curse of dimensionality to a great extent in practical cases with high dimensional data set, it is not always possible to perform clustering onto distinctive clusters using PCA.

2.4.2 Discrete Cosine Transform (DCT)

DCT is a useful approach used in signal and image processing. It transforms the signal, from the spatial domain into the frequency domain. The two-dimensional DCT is cal-

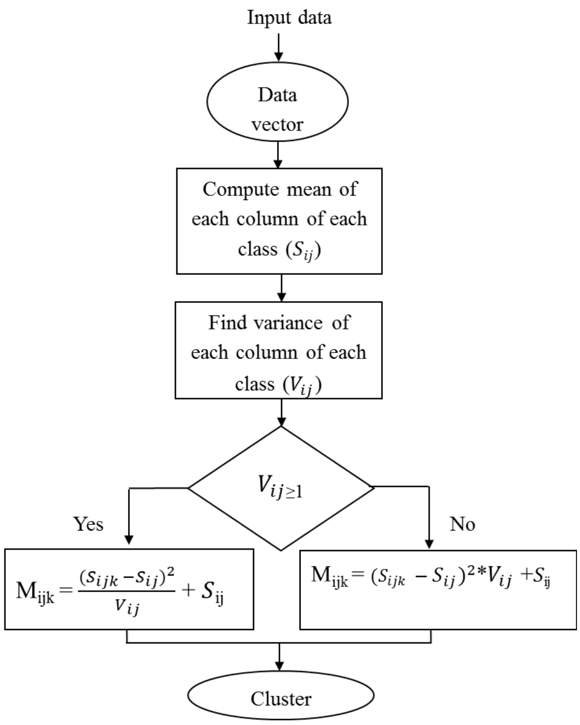

Figure 2.1: Flow chart to implement the proposed classification system.

culated by matrix multiplication to reduce the computational complexity by row-column decomposition. The mathematical model of DCT is described in [180]. The discrete cosine transform can be generalized to the unified form as follows

$$y(k) = w(k)\sum_{n=1}^{N} x(n)cos(\pi/2N(2n-1)(k-1)); \qquad k = 1, 2, \cdots, N, \tag{2.5}$$

Where $w(k) = \frac{1}{N}$ for $k = 1$ and $= \frac{2}{N}$ for $2 \leq k \leq N$; N is the length of equal-sized vectors x and y.

2.5 Mean Center Clustering

In this subsection, the algorithm of mean center clustering is explained. MCC is a clustering technique based on the first level (mean) and second level (variance) statistics of data. These statistics are very easy to calculate which makes MCC a highly efficient clustering technique. The proposed technique to center the cluster on its respective mean is described by a flow chart given in Fig. 2.1. The melody to be clustered is first represented as a data vector consisting of the samples of melody. Then the mean and variance of this data vector are calculated. Mathematically, for a data vector d containing

n samples $d_1, d_2, \cdots d_n$, the mean S_d is calculated as

$$S_d = \frac{1}{n}\sum_{i=1}^{n} d_i \tag{2.6}$$

Now, the variance V_d is calculated as

$$V_d = \frac{1}{n}\sum_{i=1}^{n} (d_i - S_d)^2 \tag{2.7}$$

Finally, the clustering is made based on the value of the variance V_d calculated in (2.7).

In order to cluster the data using MCC, the data vector d needs to be transformed into a new data vector t. Mathematically, the transformed value is calculated as

$$\begin{aligned} t =& \frac{(d_i - s_d)^2}{V_d} + S_d, \quad \text{for} \quad V_d \geq 1; \\ =& (d_i - s_d)^2 * V_d + S_d, \quad \text{for} \quad V_d < 1 \end{aligned} \tag{2.8}$$

Since each sample needs to be uniquely centered about the class mean; means of samples from each class are computed for every individual data vector. Similarly, variances are computed as an average deviation of each sample of a particular row from its class mean. The transformed values are nothing but variance normalized deviation of samples from their class mean represented about the mean itself. As a result of this transformation, data is clustered in the pattern space.

2.5.1 MCC on DCT

As a result of any preprocessing technique should be independent of the choice of training and test sets the authors were motivated to employ DCT prior to the application of MCC. It is well known that cosine transforms map the raw data in the time domain to transformed spaces in the frequency domain where the natural clusters in the data become more distinguishable. Therefore, subsequent application of MCC on DCT coefficients is likely to ensure better generalization by final classifiers. DCT coefficients obtained are clustered using MCC.

The musical data present in the audio clips consist of ordered sets of temporally varying pitch levels. Music aesthetics reduces these pitch levels into notes which forms a discernible syntax for musical compositions. In other words, some sort of clustering precedes the identification of notes and comprehension of melodic structures. This natural clustering step should be replicated in artificial classifiers for efficient classification.

In contrast, the proposed technique (MCC) clusters the data around the statistical mean of the training samples making the properties of the clusters dependent upon the choice of training and test subsets. Since the result of any preprocessing technique should be independent of the choice of training and test sets the authors were motivated to employ DCT prior to the application of MCC. It is well known that cosine transforms map the raw data in the time domain to transformed spaces in the frequency domain where the natural clusters in the data become more distinguishable. Therefore, subsequent application of MCC on DCT coefficients is likely to ensure better generalization by final classifiers. DCT coefficients obtained are clustered using MCC.

2.6 Genre Classification via MCC

Two popular classifiers: backpropagation trained Artificial Neural Network (ANN) and Support Vector Machine (SVM)are employed for the task of classification of melodic structures in this work. It is observed from the literature that the application of ANNs in music analysis in general and melody classification, in particular, has been rather limited. Nevertheless, ANNs find huge applications in the field of identification, decision making, pattern recognition, and clustering.

Neural nets are widely used in pattern recognition because of their ability to generalize and to respond to unexpected inputs/patterns. The network can be trained to perceive the criteria used to classify, and it can do so in a generalized manner allowing successful classification of new inputs that are not used during training [181]. It can work with large numbers of qualitative variables such as behaviors, provided that it can be coded, and able to use non-linear linked variables [182]. On the other hand, SVM is a popular statistically robust learning method based on risk minimization. SVM trains a classifier by finding an optimal separating hyperplane which maximizes the margin between two classes of data in the kernel induced feature space [183]. From the available literature it is observed that supervised clustering as a pre-processing step results in significant improvement in the training phase performance of ANN/SVM which may further lead to good generalization in the testing phase [184, 185]. Thus, this method essentially works for new data points with unknown labels after properly training the classifier.

To address the problem of genre classification, we utilize the proposed clustering technique followed by conventional classifiers like SVM and ANN to automatically classify music genre.

2.7 Data acquisition and description

Two different datasets are used in this work to check the validity of proposed technique. A brief description of these two datasets is given in Chapter 1, Section 1.12 (data acquisition). In the context of Indian classical music, raga is the primary melodic mode. A raga is a tonal framework for composition and characterization. Music historians have found that mediaeval European church modes bear significant similarity to ragas in Indian musical tradition [186]. A raga uses a series of melodic notes upon which a melody is built. Compared to the classification of individual melodies, automatic identification of a raga is a bit more challenging task. This is because the way the notes are approached and rendered is more important in defining a raga than the note itself. Furthermore, no two performances of the same raga need to be identical even by the same artist. Nevertheless, automatic raga identification can provide a basis for searching for similar songs and generating automated playlists that are suited for a certain aesthetic theme [76, 187].

2.8 Results and discussion

2.8.1 Results from scatter plots

The raw music files of Raga data belonging to Data set 1 are shown as a scatter plot in Fig. 2.2. Raw data here means the original unprocessed data which are music pieces belonging to different classes. The legend shows the generic name of the raga classes. For the sake of brevity, scatter plots for only the raga data set are shown. However, final classification results will be furnished both for raga and MIDI datasets.

The raw data scatter plot depicts time-domain samples obtained from the imported files. Here, D1, D2, and D3 represent dimensions with the largest, second largest and third largest variance respectively. Similar scatter plots can be drawn with other combinations of dimensions. However, for the sake of convenience, we have chosen to reproduce only Fig. 2.2 as a representative plot of raw data in the pattern space. The jumbled nature of raw data is evident from Fig. 2.2 and it is necessary to apply an appropriate preprocessing technique before subjecting it to the final classifier.

We compare the proposed clustering algorithm with some other techniques available in literature like PCA and DCT. PCA is a standard statistical technique that can be used to reduce the dimensionality of a data set of different classifiers on different musical feature sets to determine the genre of a given music piece, without much loss of information. PCA

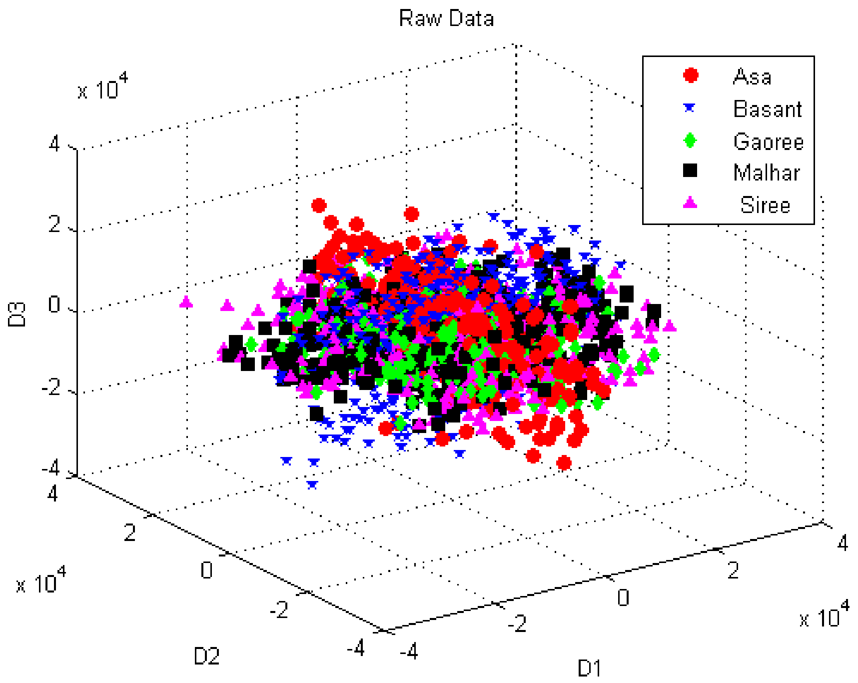

Figure 2.2: Raw data scatter plot.

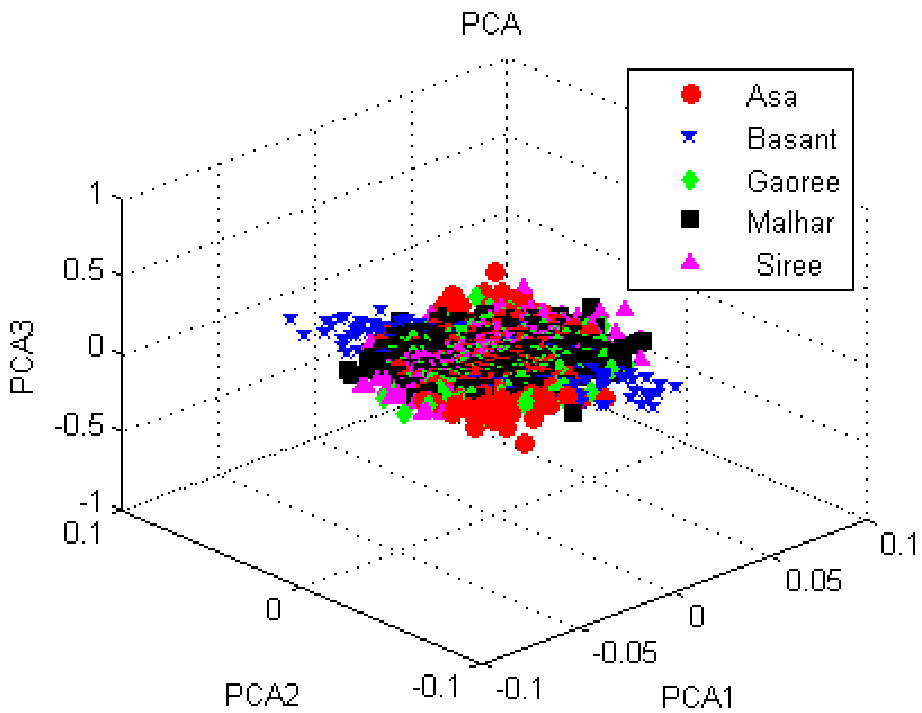

Figure 2.3: PCA scatter plot.

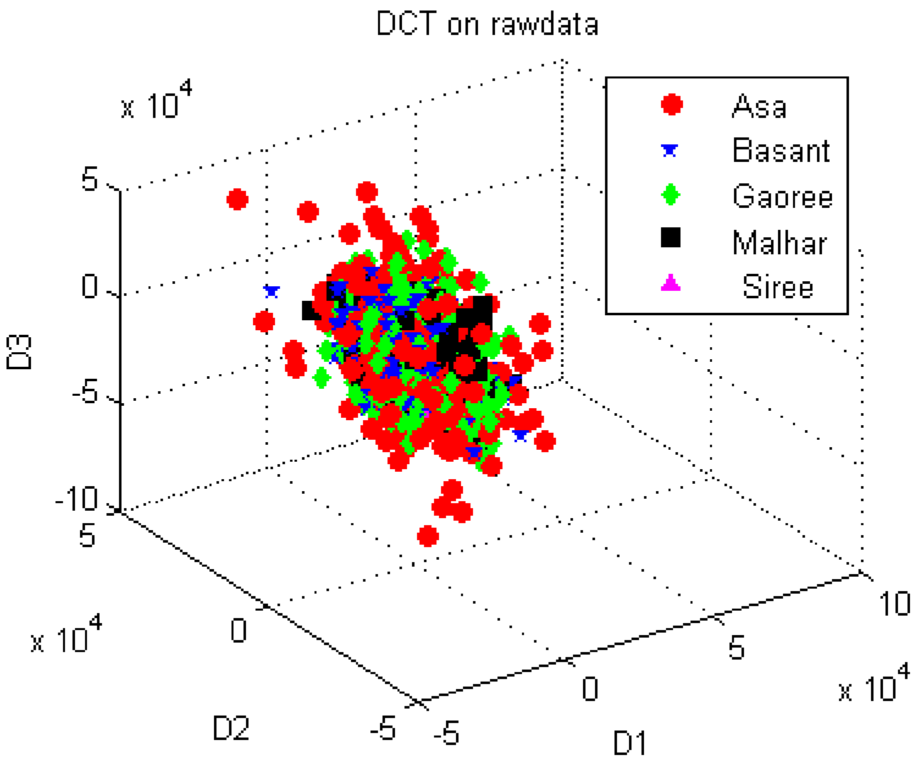

Figure 2.4: DCT scatter plot.

is an unsupervised method, which makes no use of information embodied within the class variable. Whereas, DCT is a useful approach used in signal and image processing. It transforms the signal, from the spatial domain into the frequency domain. The two-dimensional DCT is calculated by matrix multiplication to reduce the computational complexity by row-column decomposition.

The scatter plot along the first three principal component axes obtained from PCA is shown in Fig. 2.3. The scatter plot of DCT processed data is shown in Fig. 2.4. DCT and PCA produce overlapped clusters. Hence clustering efficiency is poor. The scatter plot created as a result of MCC on Raw data is shown in Fig. 2.5. It is evident from Fig. 2.5 that the application of MCC has resulted in the improved class separability in a three-dimensional pattern space. To compare the clustering results of MCC with other techniques available in literature like PCA and DCT, we utilize both scatter plots and DB index.

It is not always possible to visually analyze the class separability of the data especially when it is high dimensional. Therefore, we have employed a cluster validity measure. In this work, the DaviesBouldin (DB) index has been employed as the validity measure. This index is a function of the ratio of the sum of within-cluster scatter to between-cluster

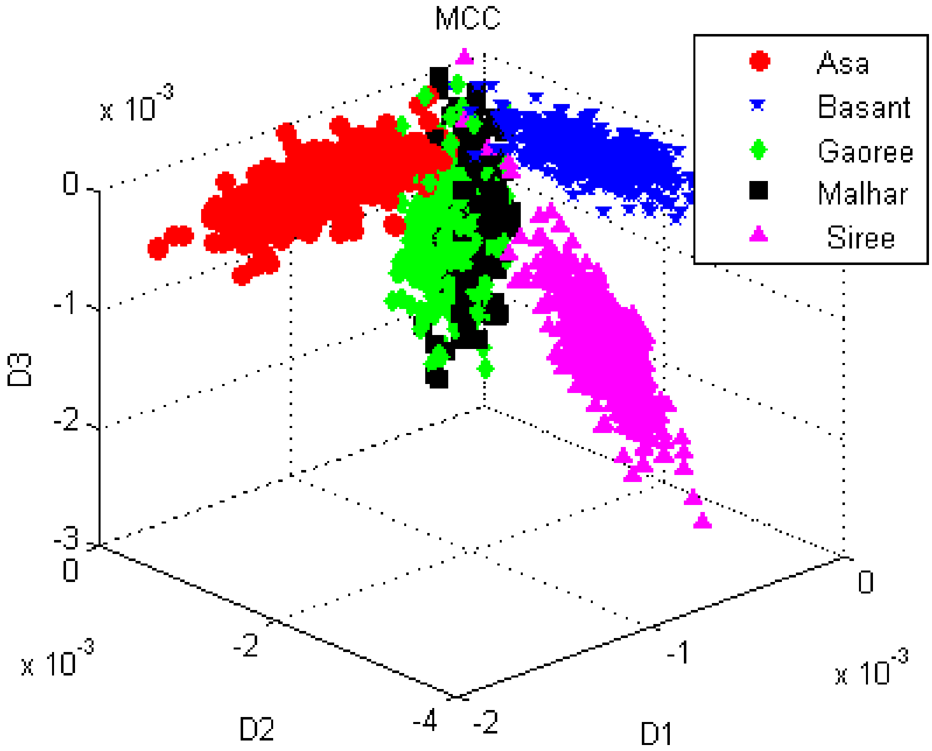

Figure 2.5: MCC on raw data scatter plot.

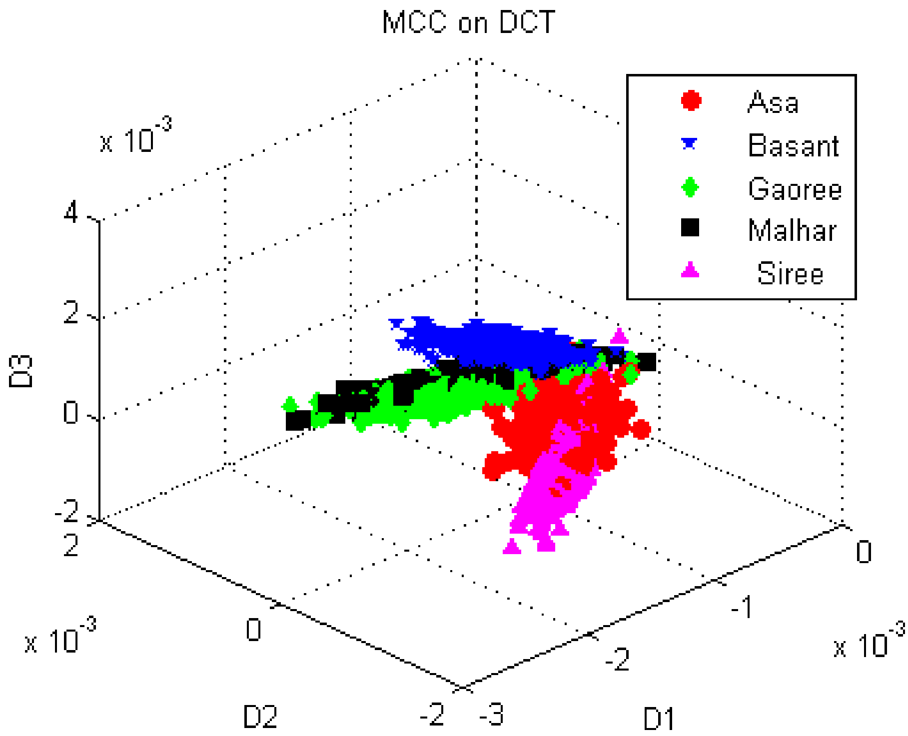

Figure 2.6: MCC on DCT scatter plot.

Table 2.1: DB indices of clusters formed with training data.

S.No.	Type of Data	DB Index (Raga data set)		DB Index (MIDI data set)	
		E.D.	M.D.	E.D.	M.D.
1	Raw data	1.29	1.06	0.98	0.76
2	PCA processed data	1.22	1.03	0.98	0.76
3	DCT coefficients	1.05	1.01	0.87	0.66
4	MCC on raw data	0.85	0.99	0.65	0.79
5	MCC on DCT coefficients	0.45	0.85	0.44	0.25

separation. A lower DB index is indicative of compact and well-separated clusters and is a measure of the effectiveness of underlying clustering technique [188].

Let σ_i be the average distance of each sample in the i^{th} cluster from its mean. If there are k clusters in the data, the DB index is given by

$$DB = \frac{1}{k}\sum_{i=1}^{k} max\left[\frac{\sigma_i + \sigma_j}{d(C_i, C_j)}\right]; \qquad i \neq j \tag{2.9}$$

Where, $d(C_i, C_j)$ is the inter-cluster distance between any two cluster centroids C_i and C_j. DB index has been calculated for both for the raw and the pre-processed data using Euclidean distance (E.D.) and Mahalanobis distance (M.D.)as metrics.

The index was computed employing both Euclidean distance and Mahalanobis distance. DB indices of clusters formed with data pre-processed using different techniques and different distance metrics have been enlisted in Table 2.1. It can be observed that MCC applied on DCT coefficients yields the best cluster validity and hence it is expected to be reflected in the final classification success rate. The MCC on DCT scatter plot is shown in Fig. 2.6. Furthermore, the raga data set has poor cluster validity as compared to MIDI data even after pre-processing. We have compared the results given in [189] in which an MFCC based classifier is used. The work is based on Carnatic Music. We have taken the cases of only the parent ragas for comparison. As given in Table 2.1 of [189], an overall accuracy of about 20 to 30% is obtained by using the Mel-frequency cepstral coefficients (MFCC) based algorithm. While the use of our proposed MCC based classifier results in 50% accuracy. Moreover, an additional 5% i.e. 55% accuracy is obtained in classification by using DCT based MCC.

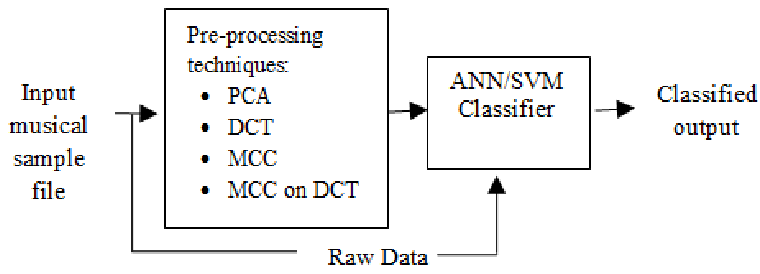

Figure 2.7: Block diagram of classification system.

It is to be noted that MCC gives unique non-overlapping clusters. DCT on MCC gives even better results. In order to prove this fact, two classifiers have been used and their performance for each case i.e. with and without MCC processed data is evaluated in the next subsection.

2.8.2 ANN and SVM Classifiers

The present work employs backpropagation algorithm trained ANN and SVM as final classifiers. Backpropagation algorithm, recognized all over the world as a powerful tool to train multilayer perceptrons, uses gradient descent technique to update the network weights [190]. PCA, DCT and MCC are used as pre-processing techniques followed by ANN/SVM as a classifier. The block diagram of classification process is shown in Fig. 2.7. It may be worth mentioning that the complexity of neural network implementing the global approximation strategy is less as compared to SVM because NN employs very small number of hidden neurons [191, 192]. On the other side the SVM is based on the local approximation strategy and uses large number of hidden units. For N number of samples, the computational cost of NN implemented by multi-layer perceptron has been found to be of the order of $N2$ where for SVM it approaches to $N3$ [193, 194].

For both SVM and ANN, 10-fold cross validation was used in this work which divides the data into a set of K subset of the same size. One of the subset is used as a validation data set in turn to test the model and remaining k-1 subsets are combined together for training data set. The cross-validation process repeats k times [195]. The classifier was trained to 500 epochs with training function trainlm being used. The number of neurons in the hidden layer was varied experimentally and a particular architecture was trained with input data a number of times. All the architectures were then simulated with the test data and actual classification performance for unlabeled samples was noted.

We have trained the network 20 times each with a particular architecture. The architecture was changed by changing the number of neurons in the hidden layer from two to

Table 2.2: ANN classification results (Raga Data).

No. of Neurons in the hidden layer	Percentage classification Pre-processing				
	Raw data	PCA	DCT	MCC on raw data	MCC on DCT
2	27.50%	31.90%	40%	53%	32%
3	35.80%	41.60%	52%	39.80%	58.90%
4	47.80%	27.50%	40%	63.50%	58.20%
5	35%	48%	47%	42.70%	67.40%
6	42%	36.60%	48.60%	42%	51.30%
7	35.50%	33.90%	39%	51%	57.50%
8	36.60%	43%	46%	47.80%	38%

Table 2.3: ANN classification results (MIDI Data).

No. of Neurons in the hidden layer	Percentage Classification Pre-processing				
	Raw data	PCA	DCT	MCC on raw data	MCC on DCT
2	33.50%	43.50%	38.20%	62.00%	53.90%
3	40.10%	40.80%	40.40%	73.80%	56.40%
4	42.70%	47.70%	53.30%	79.50%	81.30%
5	43.60%	39.90%	38.30%	74.90%	83.10%
6	49.60%	44.60%	37.60%	82.20%	78.90%
7	46.90%	46.50%	39.00%	77.10%	79.50%
8	47.00%	47.00%	39.80%	76.10%	80.60%

Table 2.4: SVM binary classification (Raga data)

Classes	Percentage Classification Pre-processing				
	Rawdata	PCA	MCC on raw data	DCT	MCC on DCT
Asa	66%	66%	99%	63%	74%
Basant	81%	81%	99%	60%	67%
Malhar	54%	58%	72%	52%	88%
Shiri	59%	61%	79%	74%	97%
Gauri	62%	54%	87%	63%	69%

Table 2.5: Confusion Matrix of Raga data (Multi SVM)

Classes	Overall percentage classification: 24.16% of test samples				
	Asa	Basant	Malhar	Shiri	Gauri
Asa	61	39	32	59	83
Basant	49	13	64	75	74
Malhar	56	14	77	54	74
Shiri	58	31	63	57	66
Gauri	76	26	23	26	124

Table 2.6: Confusion Matrix of MCC processed Raga Data (Multi SVM)

Classes	Overall percentage classification: 92.43% of test samples				
	Asa	Basant	Malhar	Shiri	Gauri
Asa	270	1	2	0	1
Basant	0	239	36	0	0
Malhar	4	48	223	0	0
Shiri	6	2	3	264	0
Gauri	0	1	0	0	274

Table 2.7: SVM binary classification (MIDI data)

Classes	Percentage classification Pre-processing				
	Raw data	PCA	DCT	MCC on raw data	MCC on DCT
Pop	69%	69%	38%	97%	76%
Folk Country	47%	50%	37%	90%	92%
Electrics	51%	53%	51%	89%	50%
Blues	55%	64%	59%	79%	76%
Alternative	41%	48%	53%	99%	94%

Table 2.8: Confusion Matrix of Multi SVM on MIDI Raw data

Classes	Overall percentage classification: 29.39 % of test samples				
	Pop	Folk Country	Electrics	Blues	Alternative
Pop	45	18	25	22	109
Folk Country	7	40	57	8	108
Electrics	6	56	49	19	90
Blues	13	44	42	25	96
Alternative	27	14	8	7	164

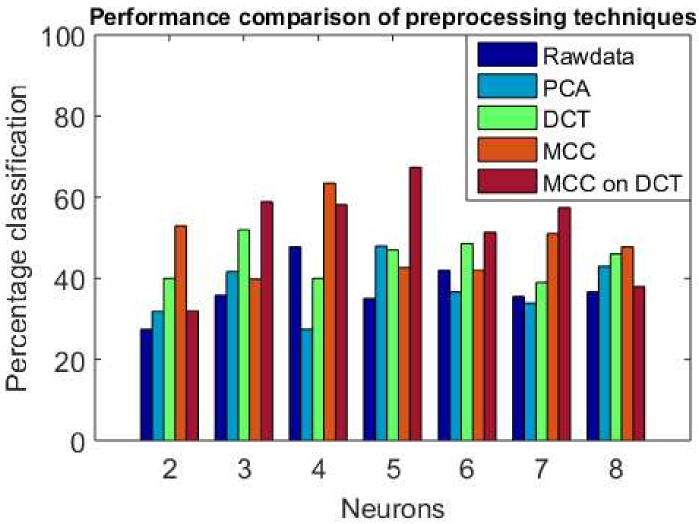

Figure 2.8: Performance comparison of preprocessing techniques on raga data (ANN classifier).

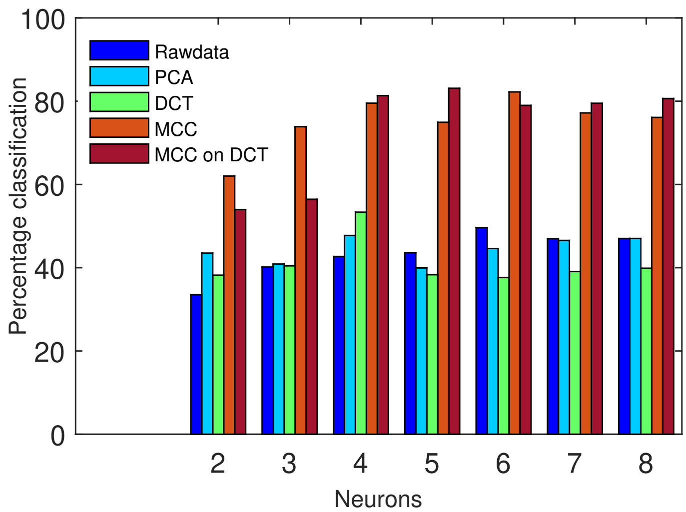

Figure 2.9: Performance comparison of preprocessing techniques on MIDI data (ANN classifier).

Table 2.9: Confusion Matrix of Multi SVM on MIDI MCC processed data

Classes	Overall percentage classification: 93.27% of test samples				
	Pop	Folk Country	Electrics	Blues	Alternative
Pop	218	0	0	1	0
Folk Country	0	194	26	0	1
Electrics	2	30	188	0	0
Blues	7	0	8	205	0
Alternative	0	0	0	0	220

eight. Since initial weights for each experiment have been saved, testing was performed for each architecture using the weight set that resulted in minimum training error. Tables 2.2 and 2.3, report testing phase performance for the ANN classifier. The melodies were taken from Raga ASA in the results given in 2.2 and 2.3 the MIDI data was taken from the western dataset of class POP. It is evident that the application of MCC on raw data has resulted in significantly better classification results for almost all the architectures. However, the best classifications were obtained when ANN was trained with MCC processed DCT coefficients. With five neurons in the hidden layer, 67.4% success rate has been obtained for the raga data.

Employing MCC on DCT has given better results with the MIDI data set where we obtained 83% success rate in the testing phase. This may be attributed to a simpler structure of features of songs in the MIDI data set. Nevertheless, the efficacy of MCC as a pre-processing stage may be deemed established. Tables 2.4 to 2.9 represent results of SVM classification both in binary and multiclass modes. Tables 2.4 and 2.6 show results in terms of percentage classification of test samples when the training samples were processed using different techniques. While tables 2.5 and 2.7 are confusion matrices showing a number of test samples correctly assigned to their respective classes. A close inspection of Tables 2.8 and 2.9 reveals that MCC processed data dramatically improves the multi-class generalization performance of the SVM classifier. The comparison of traditional techniques viz. PCA and DCT with the proposed MCC are provided in Fig. 2.8 and 2.8 for different ANN architectures. It is evident that MCC on DCT on raga data yields the best classification result with 67.4% and on MIDI data 83.1% unlabeled samples correctly assigned to their parent class. It is observed that in the case of the SVM classifier, applying DCT before MCC does not improve the performance. The reason for such results may be that DCT, in this case, is not able to provide linear separability of the data in pattern space. Instead, it makes the process more complex which results in deterioration of the results.

2.9 Conclusion

The work presented in this chapter demonstrates the efficacy of supervised clustering on the final classification of musical signals. The application of MCC on both raw data and DCT coefficients has served to improve the classification success rate substantially. Furthermore, the performance of the ANN classifier has been more consistent with the MCC processed data than with raw data, PCA processed data or with DCT coefficients. The proposed technique has given even better results with the SVM classifier. The correlation between a lower DB index of the preprocessed clusters and the percentage classification obtained using ANN/SVM classifiers underscores the necessity of cluster validity measures in the selection of an appropriate pre-processing technique. Future work may be directed towards making the pre-processing stage unsupervised so that we dont have to rely on a large pool of data for training. It is also hoped that the identification/extraction of other appropriate features from musical samples may further enhance the success rate of the final classifier.

Chapter 3

n-Gram based Template Matching for Melody Classification

This chapter reports the application of n-Gram based template matching for raga and composer based classification. In the fields of computational linguistics and probability, an n-Gram is a contiguous sequence of n items from a given sequence of text or speech. n-Gram is a very popular technique among fraternity researching in the fields of linguistics and probability which is defined as the neighboring sequence of n items obtained from a given sequence of speech, music or textual data. The type of information used in n-Gram usually depends on the application. In music analysis, n-Gram is used to calculate the similarity of a given string of text or speech corpus with data. Depending on the value of n used in an n-Gram, it is referred to as unigram, bigram, trigram or more commonly by the value of n like two-gram, three-gram, four-gram etc.

Traditionally, n-Gram is used in the problems of word prediction. The upcoming word can be predicted from the previous (N-1) words by using n-Gram. These models are also called language models or LMs. These LMs can be used to assign a probability of the next word. Estimators like the n-Gram that assign a conditional probability to possible next words can be used to assign a joint probability to an entire sentence. n-Grams models are one of the most important tools in speech and language processing. n-Gram is essential in any task in which the words must be identified from ambiguous and noisy inputs.

3.1 Applications of n-Gram

Due to its various advantageous features, the n-Gram model finds its applications in music analysis, language processing, biology (biological sequence analysis), data encryption and compression, wireless communication, probability theory, Statistical Machine Translation (SMT) and many more. Some of these advantageous features are the simplicity of the implementation of the n-Gram algorithm and the ability to be scaled up to any order from small experiments. Augmentative communication [196] is another application of the n-Gram algorithm in which people who are unable to use speech or sign language to communicate are helped by using simple body movements to select words from a menu that are spoken by the system. Using n-Gram models, word prediction can be used to suggest likely words for the menu.

Some of the other areas where n-Gram is used are as follow:

- Part of speech tagging [197]
- Natural Language Generation [113]
- Word Similarity
- Authorship identification
- Sentiment Extraction
- Predictive Text Input (Cell phones)

3.2 Advantages/drawbacks

Practical systems often use n-Gram algorithms due to its low latency and robust nature to handle spontaneous speech effects such as hesitation and lapses. Due to this, n-Gram algorithms become extremely efficient and perform well. An n-Gram language model is essentially a look-up table: no probability needs to be computed at test time, except when the model backs off and then the computation is easy. Moreover, n-Grams can be converted to FSTs which means that they can be easily integrated into the efficient FST-based search engine which uses Viterbi decoding.

However, when we target large models with lots of training data and large vocabularies, it has been shown that n-Grams are not the best choice with regards to modeling accuracy. Williams et al. [25] showed that n-Gram performance quickly saturates with increased n and that an n-Gram model is unable to exploit an increase in data beyond a certain point. Essentially this means that n-Gram models can only take us so far. If we want to increase the accuracy of our speech recognizer we have to look for other models.

3.3 Challenges in melody classification

From the perspective of pattern discovery, composer classification and raga classification are quite different tasks. The former involves the extraction of composer-specific motifs through pattern discovery and the latter requires extraction of raga-specific motifs by identifying inter-opus recurrence. Once the motifs are extracted, they can be used as a query for the subsequent pattern matching stage. As will be seen later, the implementation of a pattern matching stage has its own set of challenges. Nevertheless, the final classification can be significantly improved by making pattern matching more efficient. This work is an attempt to implement a general-purpose pattern matching paradigm which extracts class information from the motifs. The technique proposed here can accomplish both composer classification and raga classification using motifs obtained via a preceding pattern discovery stage. The performance of the pattern matching technique for raga classification has also been evaluated with catch phrases (traditional motifs).

3.4 Motivation and contribution

Musical creations and performances are often complex and sophisticated and so is its processing and analysis. Recent innovations in information technologies and especially digital signal processing have brought about tremendous changes in the way music content is used and accessed. Thus, computational musicology is a vibrant field aiming to surmount innumerable challenges. Identification of individual melodic structures poses a plethora of challenges since there is inherent subjectivity in the way humans classify musical signals. Every musical phrase possesses some unique features which help humans distinguish one piece from another. These features hold in them the complete essence of both behavioral as well as signal-based properties of music [109].

However, most of the cultures around the world have their own way to classify music; how music is perceived varies from one culture to another. Nevertheless, most of the cultures around the world possess some loose templates which are invoked when listening to a rendition. The templates (called as catch phrase in this work) may or may not be predefined for a class of music but their existence cannot be ignored. In classical Indian music, these templates are termed as Pakad (catch phrase) which represents a particular class of Indian classical music. Raga is termed as a class in Indian classical music to which melodic structures may belong based on their properties. The pakad or catch phrase of a Raga can be used as an input to n-Gram algorithm to check whether a given melody belongs to that Raga or not. The melody in itself can be constructed from any number of individual notes. However, the melody can be related to its particular class with the help of catch phrase of that class.

More emphasis is given to the way notes are approached and rendered rather than the notes themselves in raga based compositions and hence any two performances of a raga are never identical even if done by the same performer. It is very difficult to identify the class of a given melody for a listener who is not from a music background. Moreover, performances are graded in terms of their degree of match with the standard template and this attribute is a rather subjective one. On the other hand, in western classical music, this makes the scope of n-Gram based matching technique rather limited. In a nutshell, this work seeks to answer the following questions related to the classification of melodic structures in general and Indian ragas in particular.

- How to impart the ability of automatic music classification to a machine?
- Is it easier to classify those structures for which unique templates exist?
- How to efficiently utilize n-Gram matching for those melodic structures which do not possess a template?

The above questions have been answered in the backdrop of the current state of the art which makes extensive use of n-Gram techniques for melodic analysis.

3.5 n-Gram Algorithm

In n-Gram model, the matching score for k different data strings can be calculated as

$$\Phi_{n-Gram}\left(w_k\right) = w_{k-n+1}, w_{k-n+2}, \cdots, w_{k-1} \tag{3.1}$$

where the value of n is kept between the numbers 2 and 5. The n-Gram probabilities can be estimated by means of relative frequencies as

$$P\left(w_k \mid w_{k-n+1}^{k-1}\right) = \frac{C\left(w_{k-n+1}^{k}\right)}{C\left(w_{k-n+1}^{k-1}\right)} \tag{3.2}$$

where $C(\cdot)$ indicates the count or number of times a word sequence occurs in the training data. The problem with this maximum likelihood estimate, however, is that it does not take into account data sparsity. Since (3.2) factorizes the probability of a word sequence $P(w_{N-1})$ into a product of conditional probabilities $P(w_k \mid w_{kn+1}^{k1})$, this means that a single zero or undefined conditional probability $P(w_k \mid w_{kn+1}^{k1})$ leads to a zero or undefined probability $P(w_{N-1})$ of the entire word sequence. For decades researchers tried to overcome these sparsity issues with well-known smoothing techniques such as discounting, back-off and interpolation [198]. We will not dwell on this multitude of smoothing techniques, but rather refer the interested reader to Chen and Goodman [199], who made an excellent overview in addition to proposing modified Kneser-Ney smoothing, which is commonly accepted as the best n-Gram smoothing technique. Even today, some 40 years after they were introduced, n-Grams continue to be apopular, if not the most popular, choice of language models in automatic speech recognition and other fields. This success can be mostly explained by the ease with which they can be trained and evaluated and by the fact that they can be readily integrated into speech recognition systems.

Although n-Gram features reduce the parameter space of $P(w_k \mid w_{k1})$ and consequently to some extent also the data sparsity, they completely fail to capture language phenomena that span longer distances. Moreover, even with reduced contexts, data sparsity is still an issue. For an n-Gram language model to make a reliable probability estimate of a word sequence, this sequence has to occur exactly in the training data. That is, n-Grams cannot generalize to similar words, which hamper their ability to share contexts and make full use of the available training data. Fig. 3.1 shows the catch phrase, test string and windows for calculating exact 2-Gram matching for a given illustration.

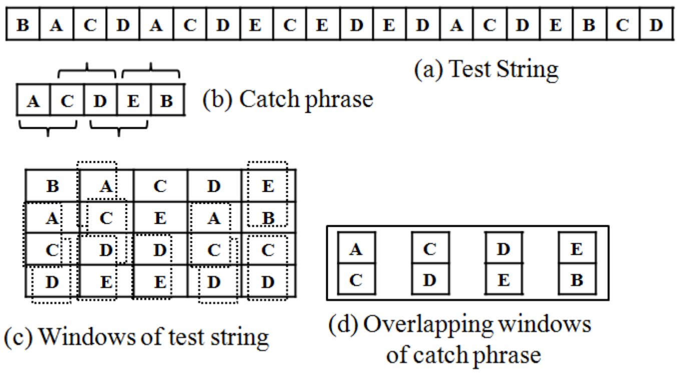

Figure 3.1: Catch Phrase, Test string and windows for calculating Exact 2-Gram matching.

3.6 n-Grams Extraction

n-Gram is simply n consecutive letters or words. There are word and character n-Grams. They overlap, i.e. each token belongs to n n-Grams. For instance, in text Music we have 3 character 3-grams: Mus, usi and sic. n-Grams are very useful in natural language processing (NLP) in the situations, where not only words are significant, e.g. in authorship attribution, language recognition or where it is hard to separate the words. A good example is Thai, where there are no whitespaces. In this aspect, Thai is especially similar to music for us it is just a flow of characters without order or semantics; however it still remains a natural language for Thais. If the NLP tools may also be applied to this language, why they cannot be applied to music as well (treated as a natural language).

The first step of n-Gram extraction after simplifying the data from MIDI files (i.e. making linear order of notes in each track), is to find what could represent unigrams. The simplest approach would be just getting the duration or pitch as the basic feature, but this does not bring good results. The pieces can be played at different speeds and can be transposed to any key. The features one needs have to be key independent so that not the absolute note pitch is important, but the relative pitch to other notes. It is crucial because the key does not tell us anything about a certain work, e.g. J. S. Bach wrote two sets of preludes and fugues, each fugue in each existing key in well-tempered scale, thus if one does the pitch distribution analysis, we will obtain one flat, normalized. The second important feature of musical n-Grams is that they should be tempo-independent. In MIDI files the duration is not given symbolically as quarters, eights, half-notes, but in a direct way, that can be mapped to milliseconds. Every MIDI file representing the same piece, but sequenced by different people (or programs) will look a little bit different. The following steps were applied:

- Calculate the similarity scores (n-Gram scores) for each composer.
- Sum up all the similarity scores of each individual composer.
- Sort the sum of scores for different composers in descending order.
- Take the composer with the highest sum as a result (classified class).

Fig. 3.2 shows a string of 12 notes which has been obtained by annotating the first clipping from the set of song samples. Equal sized overlapping windows with 4 notes have been shown. Similarly, Fig. 3.3 depicts a string of 8 notes corresponding to a catch phrase. The string of the catch phrase has four consecutive 2-grams as shown in Fig. 3.3. For n-Gram matching, each of the consecutive n-Grams of the catch phrase is searched in all overlapping windows of size 2n in the test string. Thus, initially the first 2-gram $(E, D\#)$ will be searched in the first window of the test string $(A, B, C\#, D)$ followed by a search in the 2nd, 3rd, and 4th window and so on. Any exact match yields scores a 1 and individual scores are cumulatively added for all the 2-grams. The catch phrase of a melody class, the test string to identify the corresponding class of music piece and the windows of test string for n-Gram for finding a match is shown in Fig. 3.4.

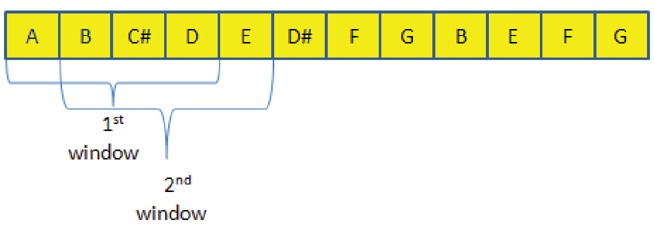

Figure 3.2: Test string with 12 notes.

Table 3.1: 2-Gram Matching Scores

Test String	Catch Phrase			
	C 1	C 2	C 3	C 4
S1	5	6	2	3
S2	11	8	2	3
S3	5	5	3	3
S4	9	6	2	4
S5	5	9	6	5
S6	5	6	7	6
S7	5	6	8	4
S8	7	8	9	5
S9	4	12	9	6
S10	5	9	8	8
S11	7	8	7	10
S12	5	12	5	3
S13	5	8	9	8
S14	6	8	6	6
S15	6	8	9	8
S16	6	6	7	5

Table 3.2: 3-Gram Matching Scores

Test String	Catch Phrase			
	C 1	C 2	C 3	C 4
S1	1	0	0	0
S2	1	0	0	0
S3	0	0	0	1
S4	2	3	0	0
S5	0	0	2	1
S6	1	1	2	1
S7	1	0	1	1
S8	0	2	0	0
S9	0	0	2	0
S10	0	1	1	0
S11	1	0	0	2
S12	1	0	0	1
S13	2	2	2	0
S14	2	0	1	0
S15	2	1	0	1
S16	1	0	2	2

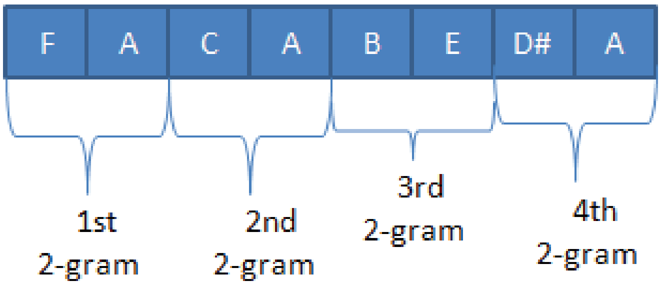

Figure 3.3: Catch phrase with 8 notes.

C	D	E	F	G	A	B	C	C	B

(a) Catch Phrase

C	A	G	F	G	E	F	G	G	F

(b) Test String

C A G F	C A G F G E	C A G F G E F G	C A G F G E F G G F
2-Gram	3-Gram	4-Gram	5-Gram

(c) Windows of Test String for n-Gram

Figure 3.4: Figure shows (a) catch phrase of a melody class (b) test string to identify the corresponding class of music piece and (c) windows of test string for n-Gram for finding a match.

As the first step n-Gram matching has been implemented for 2, 3, and 4-gram of the catch phrase. Final scores for 2, 3, and 4-gram matching have complied in the form of Tables 3.1, 3.2, and 3.3 respectively. In this work, the test string $S_1, S_2, \cdots S_{16}$ are the strings selected from Dataset-1. These results are calculated by using the datasets explained in Chapter 1. The two datasets are the classical western dataset called as Dataset 1 and classical Indian dataset called as Dataset 2.

It is evident from Tables 3.2 and 3.3 that as the length of the gram increases there are fewer occurrences of matching of the catch phrases with the test strings. This is expected since the possibility of exactly matching a longer string of the catch phrase is less than that for a shorter string of the same catch phrase. However, it was felt that during the matching of higher grams, non-exact matching should also be reflected in the final scores. Intuitively, if an exact match gives a perfect impression to a listener, an almost exact match often gives a flavor of the intended class. Therefore, we have identified matching instances of 2 and 3 notes in 3-gram and 4-gram matching respectively and modified the scores of Tables 3.2 and 3.3 adding a value of 0.33 and 0.167 for three and two note non exact matching obtained respectively while looking for a four gram match.

Table 3.3: 4-Gram Matching Scores

Test String	Catch Phrase			
	C 1	C 2	C 3	C 4
S1	0	0	0	1
S2	1	0	0	0
S3	0	0	1	0
S4	1	0	0	0
S5	0	0	0	0
S6	0	0	1	0
S7	1	0	0	1
S8	2	0	1	0
S9	0	1	0	1
S10	0	0	1	0
S11	1	0	0	0
S12	0	1	1	0
S13	0	0	1	0
S14	0	1	0	1
S15	0	0	1	0
S16	0	1	1	0

3.7 Catch phrase for string matching

Cultures around the world have their own distinct melodic structures and definition of rules to compose music. Yet they all are similar in the sense that most of them contain discernible patterns that enables a listener to classify the musical piece by composer, style, genre, geographic region, etc. However, subsequent processing of this information for the recognition and classification of melodic structures is a rather complex process. Melodic structures are identified by humans based on the arrangement of frequencies in them as well as other parameters such as tone, pitch, rhythm and tempo. Melodic structures are usually identified by matching a distinctive motif fixed in the memory to the piece of melody being played [126].

In some musical traditions, stylistic motifs are traditionally available as strings of individual notes. For instance, Indian classical music makes use of the raga framework quite extensively. The nature of raga lies between a modal scale and a tune. It consists of five to nine musical notes based on which melody is constructed [200]. Traditionally available motifs known as Pakads are associated with many ragas. The term Pakad can be roughly translated to catch phrase. The catch phrase holds the essence of a particular raga, and its presence in a recital enables the listener to identify it. Henceforth, the term catch phrase is used to describe traditionally available raga-specific motifs. Melodic frameworks similar to ragas have also been identified in other musical traditions [201]. One particular genre of traditional Chinese music consists of a repository of traditional melodies, together known as Qupai in which embellishments and variations in tempo are permissible to a great extent. The catch phrases play similar role in classification of ragas as do stylistic motifs in composer classification.

3.8 Patten Discovery Algorithm

In western classical music, no catch phrase is available for a given class of melodies. So, in order to classify the western classical music by n-Gram algorithm, a pattern discovery technique based on Structure Induction Algorithm (SIA) is used. The catch phrase is extracted from a given melody by SIA algorithm and is further used by n-Gram algorithm to calculate matching scores. The input to our melody classification algorithm is a MIDI file containing notes. For Indian classical music, the catch phrases of all the possible classes to which the given test song may belong are available. Hence, n-Gram matching algorithm can directly be applied on a test string using the given catch phrase in this case. But, in the case of western classical music, no such predefined catch phrases are available in the literature based on which classification decision could be made. Hence, a pattern discovery technique called Structure Induction Algorithm (SIA) is used to calculate the catch phrase for a test string in this work.

SIA was originally introduced by Meredith et al. [202] from which the details of this technique can be found. An advance SIA based algorithm called SIARCT [123] is used here which computes the maximal repeated patterns (MTP) by SIA (sorted into lexicographical order) and trawling inside this MTP from beginning to end, returning subsets that have compactness greater than some predefined threshold. This pattern discovery algorithm acts as a preprocessing technique to the n-Gram matching algorithm to calculate the catch phrase of a given test string. Even though the catch phrase for some melodic classes (like ragas) is available but for the sake of uniformity, all the catch phrases used in this work are generated by the pattern discovery algorithm SIARCT.

3.9 Music classification using n-Gram

Music retrieval is often treated as a string matching problem [203]. One such popular technique is the exact n-Gram matching technique which is used in this work to extract similarity between a catch phrase and test string [196]. n-Gram is created by taking a non-overlapping frame of N notes from the catch phrase and overlapping 2N notes from the test string. These frames can take any length depending on the value of N and contains adjoining notes. For n-Gram matching, each of the consecutive n-Grams of the catch phrase is searched in all windows of size $2N$ in the test string. Any exact match produces a score 1 and individual scores are cumulatively added to get the total n-Gram matching score.

An example is given in Fig. 3.4 for a better understanding of how the test string and catch phrase are divided into windows for n-Gram calculation. Fig. 3.4(a) represents a catch phrase and 3.4(b) represents a test string of 10 notes from our database. Windows of length twice the n- Gram for 2, 3, 4 and 5-Gram are shown in 3.4(c). It is to be noted that the windows of test string for a particular n-Gram are shifted without overlapping while in the case of catch phrase, these windows of length n are shifted by overlapping the last n1 notes of the previous window. Table 3.4 shows the confusion matrix of exact n-Gram showing the number of melodies classified against different classes for Dataset-1. The confusion matrices for exact n-Gram are shown in Tables 3.4 and 3.6 for Datasets-1 and 2, respectively.

Table 3.4: Confusion matrix of exact n-Gram showing number of melodies classified against different classes for Dataset-1.

		Predicted Class									
		C1	C2	C3	C4	C5	C6	C7	C8	C9	C10
Actual Class	C1	**4**	1	0	2	0	1	0	1	1	0
	C2	1	**5**	1	0	1	1	0	0	1	0
	C3	0	1	**6**	0	1	1	0	0	0	1
	C4	2	0	0	**6**	0	0	1	0	0	1
	C5	1	1	0	1	**5**	0	2	0	0	0
	C6	0	0	1	0	0	**7**	0	0	1	1
	C7	1	0	1	0	2	0	**4**	0	1	1
	C8	1	0	2	0	3	0	1	**3**	0	0
	C9	1	1	0	1	0	2	0	0	**4**	1
	C10	2	0	1	1	0	1	0	1	0	**4**
		Precision rate						**49.08%**			
		Recall rate						**48%**			
		True negative rate						**50.73%**			

3.10 Fuzzy Inference System

Fuzzy Inference System is employed in this work which utilizes fuzzy set theory to map features of input to output classes. This mapping includes fuzzifying the inputs using appropriate membership functions and determining a set of fuzzy rules to combine the fuzzified inputs. Finally the output is defuzzified to get the classification results. The FIS is implemented using Mamdani type fuzzy model. Mamdani method is widely accepted and famous among researchers for capturing expert knowledge and allows describing the expertise in more intuitive and practical manner [198]. Let us take n MIDI files $S_1, S_2, S_3, \cdots S_n$. The catch phrase calculated form these MIDI files is $C_1, C_2, C_3, \cdots C_n$. Let $\mathbf{M_1}$ is the overall matching score for a given MIDI and $M_{p,q}$ is the matching score of p^{th} MIDI with q^{th} catch phrase. The priority score for a particular catch phrase with a melody is $p(C_q)$. The algorithm for the proposed classification system using exact n-Gram matching is as follows:

Algorithm:

INPUT: Set of MIDI files; $\mathbf{S} = S_1, S_2, S_3, \cdots S_n$

1. Apply SIARCT on given test string to calculate catch phrase C_i
   for i **=** 1 **to** n
   $C_i = \mathbf{SIARCT}(S_i)$
   end for
   $\mathbf{C} \leftarrow C_1, C_2, C_3, \cdots C_n$

2. $\mathbf{M_1} \leftarrow 0$
   for n **=** 2 **to** 5

3. Divide C_q into non overlapping subsets of length n
   $C_q \leftarrow C_q(n, j)$; $j = 1$ to (length (C_q/n).

4. Divide S_i into windows of length $2n$
   $Si \leftarrow S_p(2n, k)$; $k = 1$ to (length $(S_p)/2n)$.

5. $Score = C_q - S_p$

6. **if sum**(Score)=0

7. $M_{p,q} = M_{p,q} + 1$

8. $\mathbf{M_1} = \{M_{p,q}\}$
   end for

9. Calculate Priority Score, $p(C_q)$ for Exact Fuzzy score

10. Output score = $\sum_i$Total Priority score(i) ; i=catch phrase

11. Matched string = index of max(Output score)

3.11 Membership Function and rule construction

After determining the feature vector for all possible pairs of test phrase and catch strings of database by n-Gram matching, FIS is constructed by designing membership functions and rules. Different membership functions are available to fuzzify the input feature vector like Triangular, Trapezoidal, Gaussian, Two-Side Gaussian, Bell-Shape, Product of Two Sigmoid, Differences between Two Sigmoid and Pi-Shape. Based on the best performance, Triangular, Trapezoidal and Gaussian type membership functions have been chosen and their performance is analyzed. Input score of each n-Gram is characterized using four membership functions denoted by weak (W), moderate (M), good (G) and excellent (E) match. Similar membership functions are taken for output variable. In Fig. 3.5, the chosen Membership functions used in FIS are given. The defuzzification, implication and inference settings used for the system are listed below.

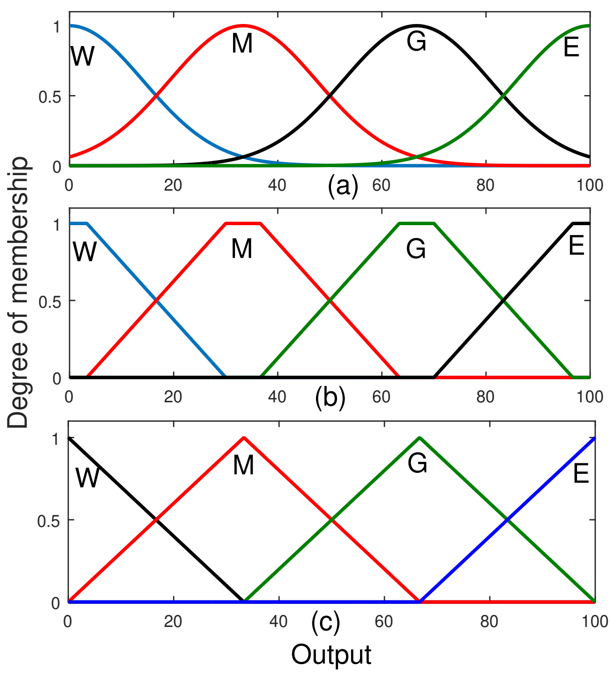

Figure 3.5: Membership functions used in fuzzy inference system.

- Centroid method is used for defuzzification and min for implication.
- Max operator is used to aggregate all the results to form the output.
- And and Or logics are implemented using Min and Max operators respectively.

The eight rules in the FIS are listed in Table 3.5. In the testing phase, when a query MIDI file (test string) is used to check the accuracy of designed FIS, the n-Gram score

Table 3.5: Rules used in the Fuzzy Inference System.

Rule No	2-Gram	3-Gram	4-Gram	5-Gram	Output	Weight
1	E	E	E	E	E	1
2	G	G	G	G	G	1
3	M	M	M	M	M	1
4	W	W	W	W	W	1
5	E	E	x	x	E	0.17
6	G	G	x	x	E	0.35
7	M	M	x	x	E	0.7
8	W	W	x	x	W	0

is calculated for given test string and is represented as feature vectors just like other test strings and catch phrases stored in database. A measure of the similarity between the feature vector of test string and a stored feature vector is evaluated and result is provided in output as a value between (0-100) indicating the similarity of test string with different catch phrases in the database. The test string is assigned to the catch phrase giving the highest matching score.

Table 3.6: Confusion matrix of fuzzy n-Gram showing number of melodies classified against different classes for Dataset-1.

		Predicted Class									
		C1	C2	C3	C4	C5	C6	C7	C8	C9	C10
Actual Class	C1	**8**	0	0	0	1	0	0	1	0	0
	C2	0	**9**	0	0	0	0	0	0	0	1
	C3	2	0	**7**	0	0	0	0	0	1	0
	C4	1	1	0	**7**	0	0	0	1	0	0
	C5	0	0	0	0	**7**	1	0	2	0	0
	C6	1	0	0	0	0	**9**	0	0	0	0
	C7	0	0	1	0	0	0	**9**	0	0	0
	C8	1	1	1	0	0	0	0	6	1	0
	C9	0	0	0	1	1	0	0	0	7	1
	C10	0	0	2	0	2	0	1	0	0	5
		Precision rate						**74.73%**			
		Recall rate						**74%**			
		True negative rate						**50.40%**			

The results given by n-Gram algorithm are further improved by using n-Gram based FIS. Confusion matrices of proposed n-Gram based FIS employing triangular, trapezoidal and Gaussian membership functions are shown in Table 3.4 to Table 3.10 for melodies from

Table 3.7: Confusion matrix of exact n-Gram showing number of melodies classified against different classes for Dataset-2.

Actual Class		Predicted Class									
		C1	C2	C3	C4	C5	C6	C7	C8	C9	C10
	C1	**5**	1	0	0	0	0	3	0	1	0
	C2	1	**3**	0	2	0	0	0	3	1	0
	C3	0	0	**6**	0	2	0	0	2	0	0
	C4	0	3	0	**4**	0	0	2	0	1	0
	C5	0	2	1	0	**3**	0	1	0	2	1
	C6	1	0	1	1	0	**5**	0	1	0	1
	C7	0	1	0	1	0	1	**3**	2	0	2
	C8	0	0	0	0	2	0	3	**4**	1	0
	C9	2	0	1	0	0	2	0	0	**4**	1
	C10	1	1	0	1	1	1	0	0	0	**5**
		Precision rate						**42.97%**			
		Recall rate						**42%**			
		True negative rate						**50.88%**			

Table 3.8: Confusion matrix of fuzzy n-Gram showing number of melodies classified against different classes for Dataset-2.

Actual Class		Predicted Class									
		C1	C2	C3	C4	C5	C6	C7	C8	C9	C10
	C1	**7**	0	1	0	0	0	0	1	0	1
	C2	1	**7**	0	0	0	0	1	0	0	1
	C3	0	1	**6**	0	0	2	0	1	0	0
	C4	0	1	0	**8**	0	0	1	0	0	0
	C5	1	0	1	0	**6**	1	1	0	0	0
	C6	0	0	0	0	1	**7**	0	0	1	1
	C7	0	0	1	0	0	0	**8**	0	0	1
	C8	0	1	0	0	1	1	0	**7**	0	0
	C9	0	0	1	1	1	0	0	0	**7**	0
	C10	1	0	0	0	0	0	0	0	1	**8**
		Precision rate						**71.41%**			
		Recall rate						**71%**			
		True negative rate						**51.00%**			

Dataset-1 and Dataset-2. The confusion matrix of exact n-Gram showing number of melodies classified against different classes for Dataset-1 is given in Table 3.4. Table 3.6 shows the confusion matrix of fuzzy n-Gram showing number of melodies classified against different classes for Dataset-1. Table 3.7 to 3.10 show corresponding results for Dataset-2.

Table 3.9: Confusion matrix of exact n-Gram for traditional motifs phrase showing number of melodies classified against different classes for Dataset-2.

		Predicted Class									
		C1	C2	C3	C4	C5	C6	C7	C8	C9	C10
Actual Class	C1	**5**	0	1	0	1	0	1	0	0	2
	C2	1	**6**	0	1	0	0	0	1	0	1
	C3	0	0	**5**	0	1	0	0	2	0	2
	C4	0	1	0	**5**	0	0	3	0	0	1
	C5	2	0	1	0	**3**	0	2	0	1	1
	C6	1	0	2	0	1	**4**	0	0	2	0
	C7	2	1	0	1	0	0	**6**	0	0	0
	C8	0	1	1	1	0	2	0	**4**	1	0
	C9	0	0	0	0	1	2	0	0	**6**	1
	C10	0	0	0	1	1	0	1	0	1	**6**
		Precision rate						**50.58%**			
		Recall rate						**50%**			
		True negative rate						**50.72%**			

Table 3.10: Confusion matrix of fuzzy n-Gram for traditional motifs showing number of melodies classified against different classes for Dataset-2.

		Predicted Class									
		C1	C2	C3	C4	C5	C6	C7	C8	C9	C10
Actual Class	C1	**7**	1	0	0	1	0	0	0	0	1
	C2	1	**7**	0	1	1	0	0	0	0	0
	C3	0	1	**8**	0	0	1	0	0	0	0
	C4	0	1	0	**7**	0	0	0	1	1	0
	C5	0	0	0	0	**8**	2	0	0	0	0
	C6	1	0	0	0	0	**9**	0	0	0	0
	C7	0	1	1	0	0	0	**8**	0	0	0
	C8	0	0	0	2	1	0	0	**7**	0	0
	C9	0	0	0	0	1	0	0	1	**7**	1
	C10	1	0	0	0	0	0	0	0	0	**9**
		Precision rate						**78.12%**			
		Recall rate						**77%**			
		True negative rate						**53.61%**			

3.12 Conclusion

The n-Gram matching algorithm which is conventionally used for the purpose of string matching has been utilized for music classification purpose. However, as is evident from the results simple matching based classification doesnt go beyond 50 % in terms of accuracy due to improvisations, embellishments and inherent variability in the composer style or the style of rendition. We also know that motifs do not come across as a whole in any rendition but appear in fragments from time to time without compromising its essence. An adept listener of vocal or instrumental music seldom misses the occurrence of the motifs even in fragmented form. This enables human listeners to identify the class of a musical piece even from the highly fragmented motifs. To train a machine to achieve the same level of sophication it is necessary to present it with exact but fragmented motifs during the training phase. The next chapter reports a similar methodology where even non- exact matches are also given scores based upon the degree of similarity of the fragmented motifs to the traditionally available motifs or the motif extracted using SIARCT.

Chapter 4

Weighted n-Gram and AHP based n-Gram pattern matching

This chapter introduces the problem of classifying a given melody into its corresponding class by using a weighted n-Gram and analytic hierarchy process (AHP) with the n-Gram technique. A weighted n-Gram based template matching algorithm is proposed to classify the classical melodies into its corresponding classes. The n-Gram matching technique is used to calculate the matching scores of a given melody. These scores are then fuzzified and a set of eight rules are formulated for constructing an AHP consisting of both exact and weighted n-Gram. The extent of matching obtained between the standard template and the test string is taken as the basis for the proposed AHP. Simulation results are provided based on the dataset explained in Chapter 1 (classical western and classical Indian). It is observed from results that the proposed method achieves better recognition accuracy than the classical n-Gram and fuzzified n-Gram based approaches explained in Chapter 3.

This chapter reports an improved pattern matching technique for composer and raga classification using a fuzzy analytical hierarchy process-based approach. The technique makes use of class-specific patterns extracted from a pattern discovery technique known as Structure Induction Algorithm for r super diagonals and compactness trawler. Further, to represent inexact matches a modified matching technique is proposed to assign weights to the exact matching scores in a probabilistic manner. Subsequently, the weighted scores are fuzzified to quantify the extent of match. Finally, the fuzzy scores are aggregated and classified on the basis of minimum Euclidean distance from an ideal solution in the pattern space. Experiments conducted on datasets containing a wide range of melodies from classical western and classical Indian background show that the proposed technique exhibits a consistently better classification success rate compared to the exact n-Gram-based approach and a widely used matching algorithm based on Levenshtein distance.

4.1 Music Classification System using AHP based n-Gram system

The Block diagram of the proposed classification system is shown in Fig. 4.1. This system consists of several blocks explained in the subsections below.

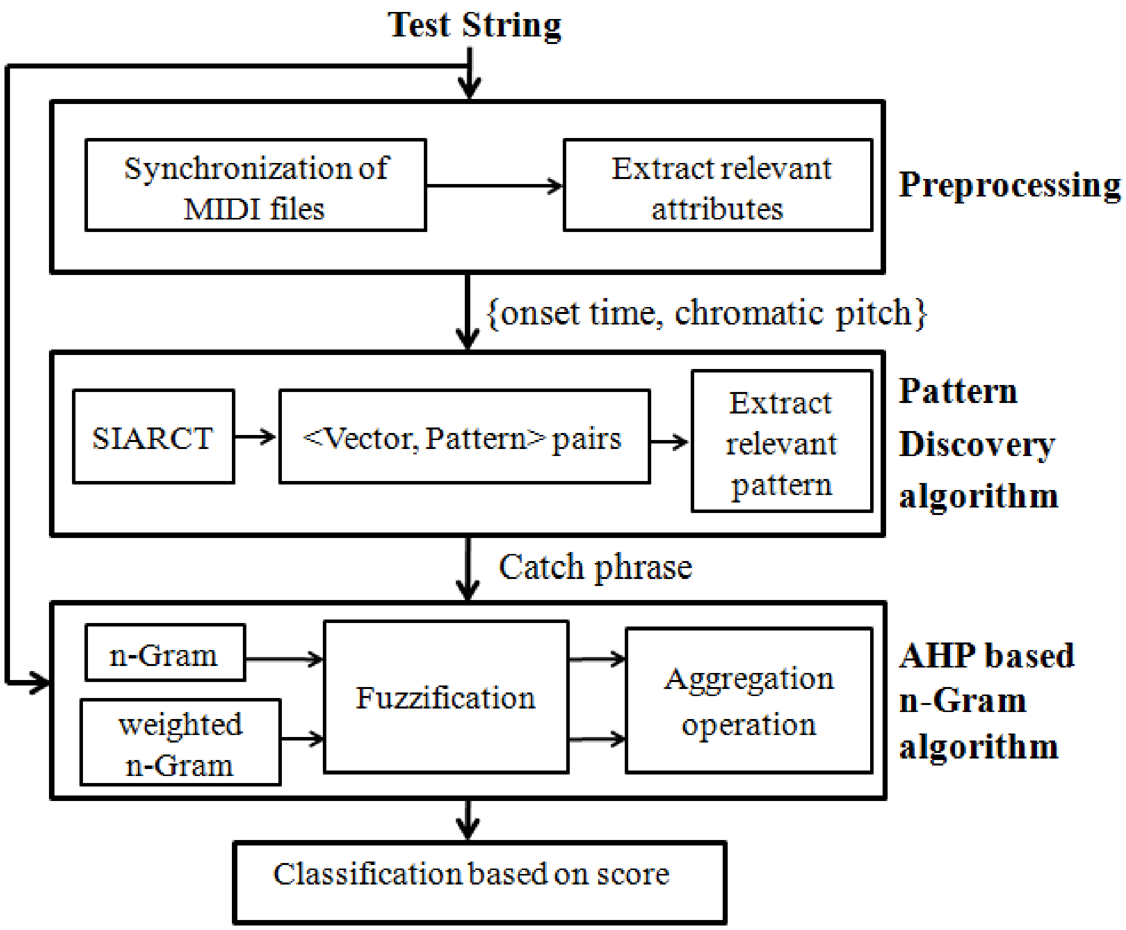

Figure 4.1: Block diagram of the systems four main blocks: Preprocessing, Pattern discovery algorithm, n-Gram algorithm and Classification.

4.1.1 Methodology

Our approach is comprised of four main blocks, as depicted in Fig. 4.1. The whole process starts with taking a test string that can belong to any one of the possible classes. No information about that test string is assumed to be known. In the first stage, preprocessing is applied to this test string. In this stage, the input files are clipped to the first thousand notes which are cross-checked by human experts. Thereafter, pattern discovery algorithm SIARCT extracts the relevant patterns from the test string which acts as the catch phrase. Finally, the AHP based n-Gram algorithm is applied to classify the given test string into its class according to its matching score. The famous n-Gram matching algorithm is utilized initially which is then improved to form a weighted n-Gram matching algorithm. Thereafter, it was observed that fuzzification of the n-Gram matching scores results in further improvement of percentage classification. Therefore, the AHP-based n-Gram algorithm includes conventional n-Gram, weighted n-Gram and fuzzification process to achieve optimum performance. Now, we present the different steps involved in this algorithm in detail in the following subsections.

4.1.2 Preprocessing

The preprocessing step is divided into two stages: synchronization of MIDI files and extraction of relevant attributes. A melody to be classified is fed as MIDI file to the algorithm. As an initial step, the MIDI file is synchronized in order to compensate offset in the global note onset time created due to error in decoding and MIDI parser. The offset in note onset time is provided with a database and cross-checked by human experts. This helps achieve tempo invariance. To maintain uniformity, each input test file was clipped to extract the first thousand note samples. This step is just a mere attempt to standardization and not compulsory. In the next step, relevant attributes are extracted from MIDI file. These attributes act as input to the pattern discovery algorithm explained in the next subsection. A standard MIDI file contains seven different attributes. These attributes of MIDI file are represented in the form of a multidimensional point set called dataset in this work. The pattern discovery algorithm works with datasets of any dimensionality. However, it will be assumed here that each dataset is a set of two-dimensional points, $< t, p >$, where t and p are, respectively, the onset time in tatums and the chromatic pitch corresponding to MIDI notes (this standard two-dimensional dataset is used in related studies) Meredith et al. [204] and Louboutin and Meredith [205]. MIDI note number is undoubtedly one of the most important attributes since it holds the core information of note. The pitch vector is directly extracted from MIDI note numbers. The information in a particular melody is represented as a series of instructions corresponding to notes that are played at specific times [206].

4.1.3 n-Gram Matching

In this section, a modified matching technique called weighted matching is explained which is employed to calculate matching score of given test string with the catch phrase. In weighted n-Gram matching technique, the methodology of creating n-Grams from catch phrase is same as exact matching as discussed in Section 3.9 but the test string (T_i)

is divided into overlapping subsets of (Nz) grams, where $z = 0, 1, \cdots, N2$. For weighted n-Gram matching, the consecutive n-Grams of the catch phrase are matched with in all windows created by (Nz) grams of test string and score is calculated like exact n-Gram case.

Weights are assigned to different n-Gram matching scores according to their importance with more weightage given to higher grams. These weights are divided linearly in the range of 0 to 1 with 1 given to the highest gram.

Using the same nomenclature for variables as defined in Section 3.10, the algorithm for the proposed classification system using exact n-Gram matching and weighted n-Gram matching in AHP is as follows:

Algorithm:

INPUT: Set of MIDI files; $\mathbf{S} = S_1, S_2, S_3, \cdots S_n$

1. Apply SIARCT on given test string
   for $n = 1$ **to** n
   $C_i = \mathbf{SIARCT}(S_i)$
   end for
   $\mathbf{C} \leftarrow C_1, C_2, C_3, \cdots C_n$
   Exact N-Gram
2. $\mathbf{M_1} \leftarrow 0$
   for $n = 2$ **to** 5
3. Divide C_q into non overlapping subsets of length n
   $C_q \leftarrow C_q(n, j)$; $j = 1$ to (length (C_q/n).
4. Divide S_i into windows of length $2n$
   $Si \leftarrow S_p(2n, k)$; $k = 1$ to (length $(S_p)/2n)$.
5. $Score = C_q - S_p$
6. **if sum**(Score)=0
7. $M_{p,q} = M_{p,q} + 1$
8. $\mathbf{M_1} = \{M_{p,q}\}$
   end for
   Weighted N-Gram
9. $\mathbf{M_2} \leftarrow 0$
   for $n = 3$ **to** 5
   for $z = 0$ **to** $n - 2$

10. Divide C_q into non overlapping subsets of $(n-z)$ grams
11. Repeat steps 7 to 10.
12. Weighted matching score
$$M_{final} = \begin{cases} 1 * M_{p,q} & z=0 \\ 0.5 * M_{p,q} & z=1 \end{cases} ; \text{ for n=3}$$
$$M_{final} = \begin{cases} 1 * M_{p,q} & z=0 \\ 0.66 * M_{p,q} & z=1 \\ 0.33 * M_{p,q} & z=2 \end{cases} ; \text{ for n=4}$$
$$M_{final} = \begin{cases} 1 * M_{p,q} & z=0 \\ 0.75 * M_{p,q} & z=1 \\ 0.50 * M_{p,q} & z=2 \\ 0.25 * M_{p,q} & z=3 \end{cases} ; \text{ for n=5}$$
end for
end for
13. $\mathbf{M_2} = \{M_{final}\}$
14. Exact Fuzzy score $= Trimf(\mathbf{M_1})$
15. Weighted Fuzzy score $= Trimf(\mathbf{M_2})$
16. Aggregate Weighted fuzzy score $= \sum_{z=0}^{n-2} \frac{\text{Weighted Fuzzy score}}{(n-1)}$
17. Calculate Priority Score, $p(C_q)$ for Exact Fuzzy score and Aggregate weighted fuzzy score
18. Total Priority score= $\mathbf{FM2}(p(C_q))$
19. Output score $= \sum_i$Total Priority score(i) ; i=catch phrase
20. Matched string = index of max(Output score)

The flow chart of algorithm for melody classification is shown in Fig. 4.2. The exact N-Gram in step 1 of algorithm is the n-Gram score without taking the weighted N-Gram score into consideration. C_q is the catch phrase extracted from q^{th} melody. The score in step 5 is calculated by using N-Gram algorithm by comparing the test string with the catch phrase which is extracted by SIARCT. The score generated by the N-Gram algorithm will be a positive integer ≥ 0. $M_{p,q}$ is a number which will give the overall exact N-Gram score for all N-Grams (2,3,4 and 5 grams in this case). Step 12 calculates the weighted matching score for each N-Gram. In order to calculate that, we have to calculate all the subsequent n-Gram scores for that n-Gram. For example, to calculate weighted matching score for 5-Gram, the n-Gram score of 2, 3, 4 and 5 gram are calculated. Similarly, for 4 gram, the n-Gram score of 2, 3 and 4 are calculated. The fuzzification is achieved by using Trimf function given in step 14 over the calculated score.

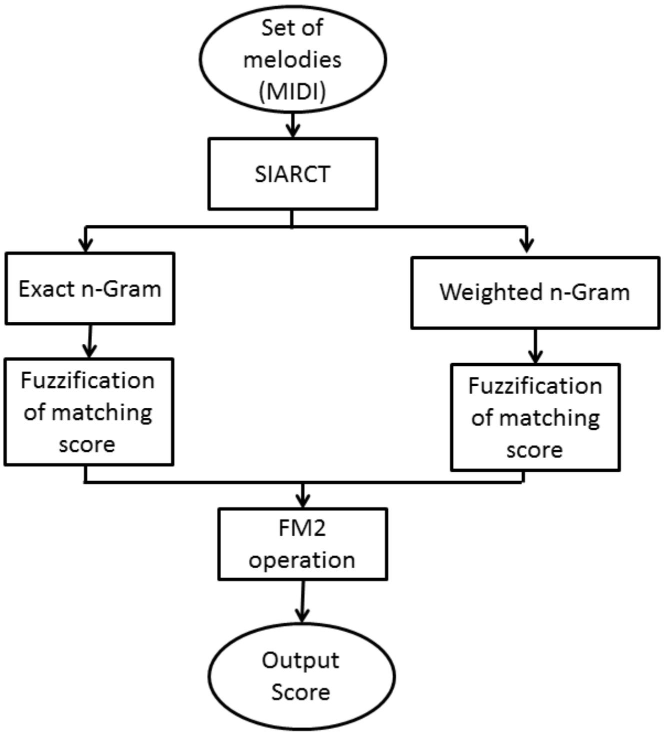

Figure 4.2: Flow chart of Algorithm for Melody Classification.

The case for n=2 is not considered in weighted n-Gram because the subsequent n-Gram does not exist in this case. Therefore, the score calculated from weighted 2-gram is equal to exact 2-gram matching score which is already taken into consideration while calculating the exact n-Gram. Therefore, n=2 case was not considered. Step 18 is used to aggregate the two matching scores i.e. exact matching and weighted matching scores using n-Gram algorithm. The motivation behind this step is that the conventional n-Gram matching is not always successful to calculate the actual matching scores. Especially in the cases when only a subpart of the string matches the catch phrase. In this case, all the subsequent n-Grams are calculated and then aggregated to give weighted n-Gram score. Both these scores give information on the extent of matching between the test string and catch phrase. Experiments were done using only one of these scores but it was observed that the collective classification percentage (exact+weighted n-Gram) was found to be much better than using either one of them. Therefore, both exact and weighted n-Grams were used in this work.

4.1.4 Fuzzified n-Gram matching

Fuzzy multiattribute decision model (MADM) is employed to classify individual strings into classes using weighted nGram matching scores as the primary metric. Initially we consider matching score for each individual gram as an independent attribute. Multiattribute decision model works well in a fuzzy framework since often there are no sharp boundaries separating one criteria from another. A particular matching score can provide much information about the extent of match. For example consistently high matching scores almost always correspond to a near-perfect match.

This motivates us to map numerical values (read matching scores) to linguistic labels (representing extent of match). In the proposed model, set of matching scores for an individual gram has been considered as an attribute with four possible linguistic labels represented by fuzzy sets. Thus, each value of n-Gram score is mapped to four fuzzy sets, i.e., Excellent Match (EM), Moderate Match (MM), Good Match (GM), and Weak Match (WM). Fuzzy memberships have been assigned to each entry employ in appropriate membership functions with relevant parameters.

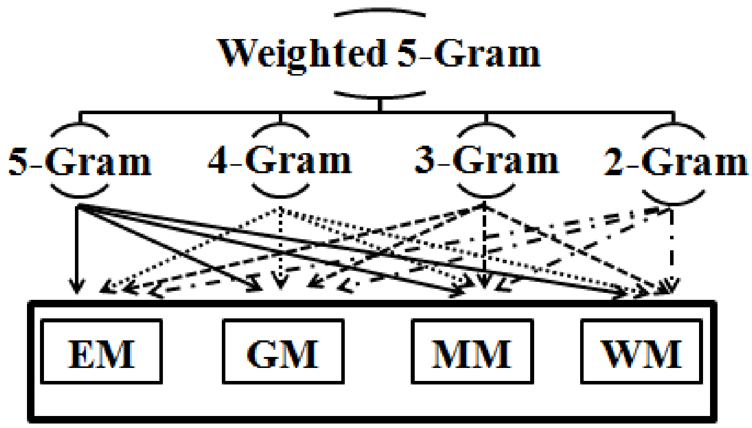

Figure 4.3: Linguistic labels of weighted 5-Gram fuzzy sets.

In the proposed model, set of matching scores for an individual gram has been considered as an attribute with four possible linguistic labels represented by fuzzy sets as shown in Fig. 4.3. Thus, each value of n-Gram score is mapped to four fuzzy sets, i.e., Excellent Match (EM), Moderate Match (MM), Good Match (GM), Weak Match (WM). Fuzzy memberships have been assigned to each entry employing appropriate membership functions with relevant parameters. The choice of membership functions and their parameters is based upon the following considerations:

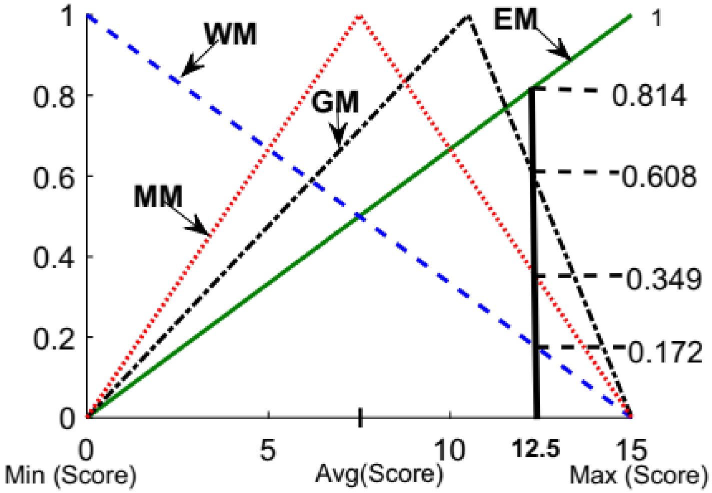

Figure 4.4: Fuzzification of a particular score of weighted 5-Gram.

- The fuzzy membership functions for individual linguistic hedges used in MADM are illustrated in Fig. 4.4. The peaks of all the four membership functions have been chosen so as to assign the highest membership value to an element on the x-axis which is quantitatively nearest to its linguistic description. For example, the highest matching score should be assigned to the linguistic set Excellent Match with a degree 1. Similarly, only those elements belong to the linguistic set Moderate Match with a high degree which lies somewhere between the maximum and minimum scores.

- The composite membership function exhibits substantial overlap between individual functions which in turn ensures that almost all the elements of the universe of discourse belong simultaneously to all the linguistic hedges with varying degrees of memberships. Similar composite functions have been constructed for other criteria as well with varying spans of the X-axis.

4.2 Aggregation of criteria

After calculating the fuzzy memberships for matching scores across all criteria in a hierarchy, the obtained values are aggregated. Thus, the final membership value of an element to the set EM is the aggregated value of the membership to the set EM across all the criteria. Similarly, the final membership values of all the elements to the remaining sets GM, MM and WM can be calculated. The aggregation operation returns a set of four elements (final memberships to the sets EM, GM, WM and MM) after the merger of the hierarchies. In this work, we have made use of FM2 fuzzy aggregation operator (an averaging type operator chosen to best fit our problem) [149]. By definition, for n values $a_1, a_2, a_3, \cdots a_n$, FM2 is calculated as:

$$FM2(a_1, a_2, a_3) = \begin{cases} min(a_1, a_2, a_3, \cdots a_n) & \Delta > \Omega \\ max(a_1, a_2, a_3, \cdots a_n) & \Delta < \Omega \\ 0.5(min(a_1, a_2, a_3, \cdots a_n) + & \\ max(a_1, a_2, a_3, \cdots a_n)) & \Delta = \Omega \end{cases} \tag{4.1}$$

where $\Delta = min((1a1), (1a2), (1a3), \cdots (1 - a_n))$, $\Omega = min(a1, a2, a3, \cdots a_n)$, while max and min are the maximum and minimum value of $(a1, a2, a3, \cdots a_n)$, respectively.

Let us take an example to explain the working of FM2 given in (4.1). Suppose we have to calculate the N-Gram score for the weighted 5-gram case. So the first step will be to calculate the 5, 4, 3 and 2-gram matching scores for this case. Let the calculated scores in the case be $S5, S4, S3$ and $S2$ respectively for 5, 4, 3 and 2-gram matching. Thereafter, these scores are to be aggregated into a single weighted score. So FM2 is used in this case as

$$FM2(S2, S3, S4, S5) = \begin{cases} min(S2, S3, S4, S5) & \Delta > \Omega \\ max(S2, S3, S4, S5) & \Delta < \Omega \\ 0.5(min(S2, S3, S4, S5) + & \\ max(S2, S3, S4, S5)) & \Delta = \Omega \end{cases} \tag{4.2}$$

The formula holds true for any number of inputs. In the present work, arguments $a1$, $a2$ and $a3$ are membership values of an element to an individual linguistic set (say EM) across any three criteria (say 2-Gram, 3-Gram and 4-Gram). The following example illustrates a remarkable property of the FM2 operator which has served as a motivation for us to employ it for aggregation.

Let 2-Gram, 3-Gram and 4-Gram matching scores be fuzzified and assigned to the set EM with the following membership values. $(EM54, EM53, EM52) = (a1, a2, a3) = (0.2, 0.2, 0.8)$. It is evident that the degree of match between the motif and the string is inconsistent across criteria. For consistently high memberships, the aggregated value as returned by FM2 is closer to the maximum value, and for consistently low memberships, it is closer to the minimum value. However, any inconsistency in the degree of memberships across the criteria yields an aggregated value closer to the arithmetic mean of the arguments. A high matching score returns a higher value of membership for the linguistic set EM. It must be evident from the composite membership function depicted that a perfect matching score will return a membership value 1 to the set EM and 0 to all other sets. This leads to the identification of an ideal solution to the classification problem.

Let any random string from the datasets be matched to any three catch phrases/motifs and aggregated fuzzy membership values are calculated. A four-dimensional pattern space can be constructed in which fuzzified matching scores can be located. A three-dimensional version of such a space is depicted in Fig. 4.5. The ideal solution is identified as a point in the pattern space which has membership value of 1 to the linguistic set EM and membership value of 0 to the rest. Since final membership values are obtained after aggregation of several criteria/subcriteria, the ideal solution can only be obtained if the matching score of the test string with a particular motif/catch phrase is a maximum in all the criteria/subcriteria of the AHP model. Such a solution is crisp and no longer remains a fuzzy set. Out of all the available alternatives, the one having a minimum Euclidean distance from the ideal solution must be chosen.

From Fig. 4.5, Euclidean distances of solutions C1, C2, and C3 from the ideal solution can be calculated as follows:

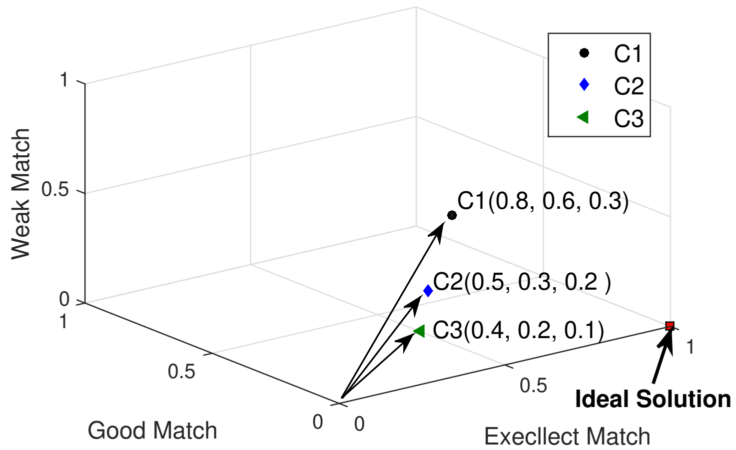

Figure 4.5: Pattern space representation of different classes C1, C2 and C3.

$$||I - C_q|| = \sqrt{((1 - x_1)^2 + (0 - x_2)^2 + (0 - x_3)^2 + (0 - x_4)^2)} \quad (4.3)$$

Where I and C_q are four-dimensional ideal solution vector and membership vector of the q^{th} alternative respectively, x_1, x_2, x_3, and x_4 are aggregated membership values to the sets EM, GM, MM and WM respectively. In Fig. 4.5, three alternatives are located on the three-dimensional plane comprising of EM, GM and WM. The set MM has been omitted to enable visualization. After the calculation of aggregated memberships, the priority score $p(Cq)$ of each alternative across all criteria is calculated by the formula.

$$p(C_q) = E(C_q) - G(C_q) - M(C_q) - W(C_q) \tag{4.4}$$

where $E(C_q)$, $G(C_q)$, $M(C_q)$ and $W(C_q)$ are aggregated values of linguistic labels EM, GM, MM and WM corresponding to catch phrase of class C_q.

It can be observed that the highest priority score always corresponds to the smallest Euclidean distance between the candidate solution and the ideal one. Moreover, calculation of priority using Eq. (11) is computationally less complex. Although alternative C3 has the smallest membership to the set EM, its Euclidean distance to the ideal solution is minimum. Thus, there is a maximum likelihood of C3 being the best alternative with the highest priority score. As discussed above, the motive behind assigning linguistic labels is to quantify any inconsistency in the magnitude of matching scores across closely related criteria. This inconsistency pulls an alternative away from the ideal solution. Therefore, a high membership to the set EM does not guarantee proximity to the ideal solution.

4.2.1 Justification for employing AHP based model

The need to employ a hierarchical model for ranking of alternatives can be understood from the following example
The catch phrase (traditional motif) for raga Yaman is:

C D E F# G A B C

Case 1: Let the first 5-Gram (C D E F# G) of the catch phrase be searched for in a test string. There is only one way to find an exact match (i.e. C D E F# G exists as a continuous sequence in the test string at least once). However, there are 3 ways to find a 3-note fragment of this particular 5-Gram (C D E: D E F#: E F# G) in the test string.
Case 2: Instead of searching for the first 5-Gram, we now search for the first 4-Gram (C D E F#) in the test string. Here also there is only one way to find an exact match (i.e. C D E F# existing as a continuous sequence in the test string). However, there are 2 ways to find a 3-note fragment of this particular 4-Gram (C D E: D E F#) in the test string.
The above cases demonstrate that matching scores for a particular criterion are dependent upon the distribution of the fragments of the motif along with the recital.

In a given test string, the probability of getting a match with a particular catch phrase/motif is inversely proportional to the length of the catch phrase/motif. On the other hand, the importance of matching a test string with a higher gram of a catch phrase is intuitively higher as compared to the matching of smaller grams since smaller fragments are more likely to be embellishments and improvisations rather than specific motifs. This contradiction can only be resolved by appropriately assigning weights to the matching of varying lengths. We propose to assign weights in a probabilistic manner with an aim to reflect the information content of a particular match. In information-theoretic terms entropy of a system is a measure of its information content and is given as

$$H(x) = -\sum_{i=1}^{X} P(x_i) log(P(x_i)) \tag{4.5}$$

In the context of our problem let us assume x to be the random variable representing a particular match and $P(x)$ be its probability mass function of x. The function $P(x)$ for an n-Gram is calculated as the probability of getting an exact match of all the notes in the motif with the notes of test string. For example, in the case of 5-Gram weighted matching, the number of notes in test string are 10. The motif is supposed to be divided into strings of 5 notes each. However, the algorithm run might yield weighted matches of 4, 3 or 2 notes which are subsets of the original 5-Gram. These notes are chosen from 128 different MIDI notes. There are six ways to exactly match a 5-note fragment of the motif in a 10 note window of the test string. Hence, for 5-Gram weighted matching, $P(x)$ for 5-Gram is $6/128^5$ for 4-Gram is $7/128^4$, for 3-Gram is $8/128^3$ and for 2-Gram is $9/128^2$. By using the values of these P(x) in (3) and after normalization, the weights for weighted 5-Gram matching are calculated as $w1 = 1.0$, $w2 = 0.7617$, $w3 = 0.5322$ and $w4 = 0.3226$. The term normalization used here is defined as $normalization(x) = (abs(log(x)))/(max(x))$; where $abs(\cdot)$ and $max(\cdot)$ are absolute and maximum operators respectively. Similarly, the weights for other weighted n-Grams can be calculated. An important point to note here is that larger weights are assigned to a higher gram catch phrase. Thus their importance is reflected in the overall weighted n-Gram matching algorithm. The weights thus generated are nothing but constituents of the entropy or overall information content of the matching scheme.

The matrix showing relative importance (priority matrix) of n-Gram weights is given as

$$C = \begin{array}{c} \\ \text{5-Gram} \\ \text{4-Gram} \\ \text{3-Gram} \\ \text{2-Gram} \end{array} \begin{array}{c} \begin{array}{cccc} \text{5-Gram} & \text{4-Gram} & \text{3-Gram} & \text{2-Gram} \end{array} \\ \begin{bmatrix} 1 & 0.7617 & 0.5322 & 0.3226 \\ 1/0.7617 & 1 & 0.7011 & 0.4274 \\ 1/0.5322 & 1/0.7011 & 1 & 0.6133 \\ 1/0.3226 & 1/0.4274 & 1/0.6133 & 1 \end{bmatrix} \end{array} \tag{4.6}$$

The priority matrix represents the relative importance of a particular criterion over another. The rows represent the length of the queries in terms of n-Grams of the motifs.

The columns represent the length of the fragment of the query which was actually found to be matching with the test string. Individual elements of the matrix are multiplying factors or weights. The matching score is multiplied with the corresponding weights to obtain the final score for a particular n-Gram. A class is assigned to a melody based on this final score as the melody is likely to belong to the class with the highest matching score. However, a possible limitation of this method becomes evident if we ponder over the following questions.

To what extent the numerical value of a matching score is a true representative of the class? Since traditional AHP evaluates several criteria which represent decisions of various experts how feasible it is to represent matching scores of individual grams as a replacement for experts knowledge? The limitations of conventional AHP become apparent when we compare weighted matching scores for two motifs/catch phrases C1 and C2. If we look at the 5-Gram criterion the test string has a higher likelihood of belonging to C2. However, according to the 4-Gram criterion, the string should be assigned to C1. It must be remembered that we have already weighted one criterion with respect to another (using the priority matrix) to obtain the scores. The most obvious method for reaching a decision is to calculate and compare cumulative scores for C1 and C2. This involves a possible loss of qualitative information. From an expert listeners point of view, the degree of "goodness" of an inexact match determines whether the motif (or its fragment) is good enough to convey the essence of its parent class (composer or raga). Since the

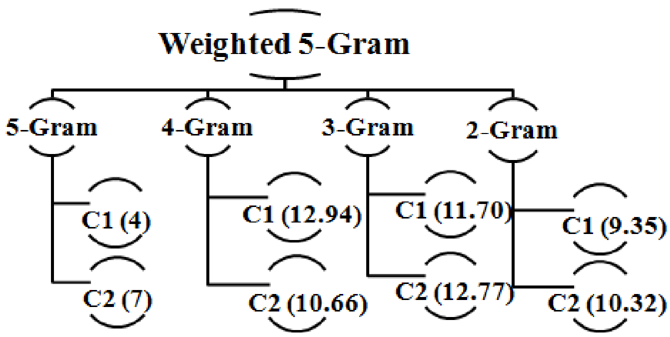

Figure 4.6: Score of different classes C1 and C2 for weighted 5-Gram.

expert's decision often comes in the form of linguistic labels, there is clearly a need to appropriately modify our model so that quantitative values are mapped to a qualitative difference. Hence, in order to further improve the classification success rate, fuzzification of the weighted n-Gram matching scores is envisaged. Fig. 4.6 shows the scores of different classes C1 and C2 for weighted 5-Gram.

4.3 Results and Discussion

The proposed algorithm was evaluated in terms of four standard metrics i.e. Precision rate, Recall rate, True negative rate, F-measure and Percentage classification as defined below

$$\text{Precision rate} = \frac{S_D}{S_D + D_D} \times 100 \tag{4.7}$$

$$\text{Recall rate} = \frac{S_D}{S_D + S_{ND}} \times 100 \tag{4.8}$$

$$\text{True negative rate} = \frac{D_{ND}}{D_{ND} + D_D} \times 100 \tag{4.9}$$

$$\text{F-score} = 2 \times \frac{\text{Precision rate} \times \text{Recall rate}}{\text{Precision rate} + \text{Recall rate}} \tag{4.10}$$

where S_D, D_D, S_{ND} and D_{ND} are the number of similar songs detected, the number of dissimilar songs detected, and the number of similar songs that are not detected and the number of dissimilar songs correctly rejected respectively. Percentage Classification is defined as the ratio of a number of songs correctly detected in a class by the total number of songs in class $\times$ 100. All these matrices can be calculated from the confusion matrix of actual class v/s predicted class for a given set of melodies from Dataset-1 and Dataset-2. In this work, for Dataset-2, two different versions are available, one with the predefined catch phrase and one without using the predefined catch phrase in which the catch phrase is calculated using SIARCT. Since our dataset consists of 10 melodies per class, each confusion matrix row must sum to 10. For the ideal case of 100% correct classification, we have a diagonal confusion matrix (all zeros, except for the diagonal elements, which should all be 10). Class names (C1 to C10) are shown for both actual classes and predicted class.

The data was first partitioned equally into training and test sets. Motifs were extracted from the training sets through the pattern discovery stage. Subsequently, n-Grams of the extracted motifs were matched with the note strings of the songs in the testing set. Five-fold cross-validation was performed to rule out overfitting. The scheme to choose the training set and testing set from given melodies is depicted in Fig. 4.7. A block of 50 songs (starting from song no. 1-50) is taken as the training set and the remaining 50 are assigned to the testing set. Subsequently, another block of 50 songs (starting from song no. 10-60) is assigned to the training in the second fold. Data is similarly partitioned in the remaining folds as well.

Table 4.10 shows the F-score for different datasets. The classification results have been

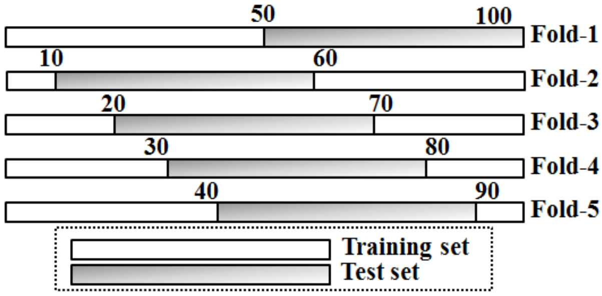

Figure 4.7: Scheme for fragmentation of data into testing and training data sets.

Table 4.1: Confusion matrix of exact n-Gram showing number of melodies classified against different classes for Dataset-1.

		Predicted Class									
		C1	C2	C3	C4	C5	C6	C7	C8	C9	C10
Actual Class	C1	**4**	1	0	2	0	1	0	1	1	0
	C2	1	**5**	1	0	1	1	0	0	1	0
	C3	0	1	**6**	0	1	1	0	0	0	1
	C4	2	0	0	**6**	0	0	1	0	0	1
	C5	1	1	0	1	**5**	0	2	0	0	0
	C6	0	0	1	0	0	**7**	0	0	1	1
	C7	1	0	1	0	2	0	**4**	0	1	1
	C8	1	0	2	0	3	0	1	**3**	0	0
	C9	1	1	0	1	0	2	0	0	**4**	1
	C10	2	0	1	1	0	1	0	1	0	**4**
		Precision rate						**49.08%**			
		Recall rate						**48%**			
		True negative rate						**50.73%**			

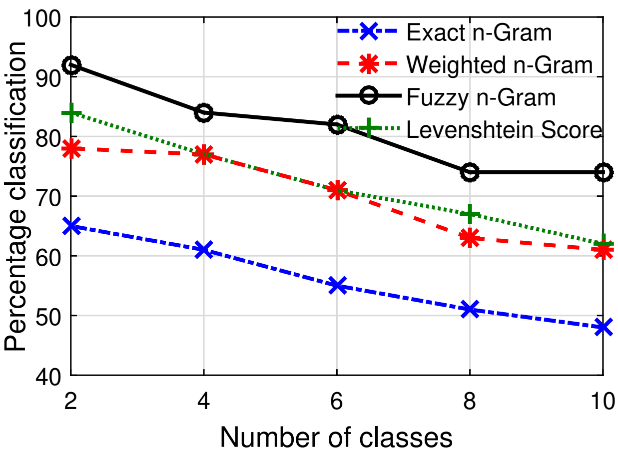

Figure 4.8: Percentage classification results v/s number of classes for Dataset-1.

shown in Tables 4.1 to 4.9 in the form of confusion matrices. Number of songs correctly classified as shown in the tables is rounded off value of arithmetic average of the number

Table 4.2: Confusion matrix of weighted n-Gram showing number of melodies classified against different classes for Dataset-1.

		Predicted Class									
		C1	C2	C3	C4	C5	C6	C7	C8	C9	C10
Actual Class	C1	**6**	0	2	0	1	0	0	0	0	1
	C2	0	**6**	1	1	0	0	0	0	1	1
	C3	1	0	**7**	0	0	1	0	0	1	0
	C4	1	0	1	**5**	0	1	0	2	0	0
	C5	0	1	0	0	**7**	0	1	0	0	1
	C6	2	0	1	0	0	**7**	0	0	0	0
	C7	0	0	3	0	0	2	**5**	0	0	0
	C8	0	2	0	0	1	0	0	**5**	1	1
	C9	0	0	0	0	0	1	0	0	**7**	2
	C10	1	1	0	0	0	0	0	0	2	**6**
		Precision rate						**64.37%**			
		Recall rate						**61%**			
		True negative rate						**53.48%**			

Table 4.3: Confusion matrix of fuzzy n-Gram showing number of melodies classified against different classes for Dataset-1.

		Predicted Class									
		C1	C2	C3	C4	C5	C6	C7	C8	C9	C10
Actual Class	C1	**8**	0	0	0	1	0	0	1	0	0
	C2	0	**9**	0	0	0	0	0	0	0	1
	C3	2	0	**7**	0	0	0	0	0	1	0
	C4	1	1	0	**7**	0	0	0	1	0	0
	C5	0	0	0	0	**7**	1	0	2	0	0
	C6	1	0	0	0	0	**9**	0	0	0	0
	C7	0	0	1	0	0	0	**9**	0	0	0
	C8	1	1	1	0	0	0	0	6	1	0
	C9	0	0	0	1	1	0	0	0	7	1
	C10	0	0	2	0	2	0	1	0	0	5
		Precision rate						**74.73%**			
		Recall rate						**74%**			
		True negative rate						**50.40%**			

of correct classification over five folds. The confusion matrices containing number of melodies classified with exact n-Gram matching algorithm are given in Tables 4.2 and 4.5 for Datasets-1 and 2 respectively. The confusion matrices for weighted n-Gram are shown in Tables 4.3 and Tables 4.6 for Datasets-1 and 2 respectively.

The confusion matrices for exact n-Gram have been shown in Tables 4.4 and 4.7 for Datasets-1 and 2 respectively. The advantage of fuzzification is evident from these ta-

Table 4.4: Confusion matrix of exact n-Gram showing number of melodies classified against different classes for Dataset-2.

		Predicted Class									
		C1	C2	C3	C4	C5	C6	C7	C8	C9	C10
Actual Class	C1	**5**	1	0	0	0	0	3	0	1	0
	C2	1	**3**	0	2	0	0	0	3	1	0
	C3	0	0	**6**	0	2	0	0	2	0	0
	C4	0	3	0	**4**	0	0	2	0	1	0
	C5	0	2	1	0	**3**	0	1	0	2	1
	C6	1	0	1	1	0	**5**	0	1	0	1
	C7	0	1	0	1	0	1	**3**	2	0	2
	C8	0	0	0	0	2	0	3	**4**	1	0
	C9	2	0	1	0	0	2	0	0	**4**	1
	C10	1	1	0	1	1	1	0	0	0	**5**
		Precision rate						**42.97%**			
		Recall rate						**42%**			
		True negative rate						**50.88%**			

Table 4.5: Confusion matrix of weighted n-Gram showing number of melodies classified against different classes for Dataset-2.

		Predicted Class									
		C1	C2	C3	C4	C5	C6	C7	C8	C9	C10
Actual Class	C1	**4**	1	0	0	2	0	0	1	0	2
	C2	2	**3**	1	0	1	0	3	0	0	0
	C3	0	1	**5**	1	0	1	0	0	1	1
	C4	1	0	1	**3**	1	0	1	2	0	1
	C5	0	1	2	0	**4**	0	1	0	1	1
	C6	1	0	1	0	0	**6**	0	0	0	2
	C7	0	0	0	0	0	0	**7**	1	0	2
	C8	0	0	0	1	0	2	0	**6**	0	1
	C9	2	0	1	0	0	0	0	0	**7**	0
	C10	0	1	0	0	1	1	1	0	0	**6**
		Precision rate						**52.18%**			
		Recall rate						**51%**			
		True negative rate						**51.33%**			

bles. Using fuzzy n-Gram in Dataset-1, 74% of the samples were correctly identified as compared to 61% when weighted n-Gram was employed. A significant increase of 10.36% in precision rate and 13 % in recall rate was observed for Dataset-1 while comparing the performance of weighted n-Gram and fuzzified n-Gram algorithms.

It is also observed that fuzzification improved the results using Dataset-2 where 71% samples have been correctly classified as compared to 51% when weighted n-Gram was

Table 4.6: Confusion matrix of fuzzy n-Gram showing number of melodies classified against different classes for Dataset-2.

		Predicted Class									
		C1	C2	C3	C4	C5	C6	C7	C8	C9	C10
Actual Class	C1	**7**	0	1	0	0	0	0	1	0	1
	C2	1	**7**	0	0	0	0	1	0	0	1
	C3	0	1	**6**	0	0	2	0	1	0	0
	C4	0	1	0	**8**	0	0	1	0	0	0
	C5	1	0	1	0	**6**	1	1	0	0	0
	C6	0	0	0	0	1	**7**	0	0	1	1
	C7	0	0	1	0	0	0	**8**	0	0	1
	C8	0	1	0	0	1	1	0	**7**	0	0
	C9	0	0	1	1	1	0	0	0	**7**	0
	C10	1	0	0	0	0	0	0	0	1	**8**
		Precision rate						**71.41%**			
		Recall rate						**71%**			
		True negative rate						**51.00%**			

Table 4.7: Confusion matrix of exact n-Gram for traditional motifs phrase showing number of melodies classified against different classes for Dataset-2.

		Predicted Class									
		C1	C2	C3	C4	C5	C6	C7	C8	C9	C10
Actual Class	C1	**5**	0	1	0	1	0	1	0	0	2
	C2	1	**6**	0	1	0	0	0	1	0	1
	C3	0	0	**5**	0	1	0	0	2	0	2
	C4	0	1	0	**5**	0	0	3	0	0	1
	C5	2	0	1	0	**3**	0	2	0	1	1
	C6	1	0	2	0	1	**4**	0	0	2	0
	C7	2	1	0	1	0	0	**6**	0	0	0
	C8	0	1	1	1	0	2	0	**4**	1	0
	C9	0	0	0	0	1	2	0	0	**6**	1
	C10	0	0	0	1	1	0	1	0	1	**6**
		Precision rate						**50.58%**			
		Recall rate						**50%**			
		True negative rate						**50.72%**			

employed. The gain in precision rate has been 19 % and that in recall rate has been 20%. Traditional motifs are available in literature for Dataset-2. We have applied the proposed technique using traditional motifs and the obtained results have been compared with those obtained when motifs were extracted through SIARCT. Confusion matrices for exact, weighted and fuzzy n-Gram using Dataset-2 and traditional motifs have been shown in Tables 4.7, 4.8 and 4.9 respectively. It is evident that the proposed weighting scheme and subsequent Fuzzification has significantly improved the classification performance,

Table 4.8: Confusion matrix of weighted n-Gram for traditional motifs phrase showing number of melodies classified against different classes for Dataset-2.

		Predicted Class									
		C1	C2	C3	C4	C5	C6	C7	C8	C9	C10
Actual Class	C1	**7**	1	0	0	0	0	0	1	0	1
	C2	0	**8**	0	0	0	0	0	0	1	1
	C3	0	0	**7**	0	0	0	0	1	1	1
	C4	1	0	0	**6**	1	1	1	0	0	0
	C5	1	0	0	0	**8**	0	0	0	1	0
	C6	0	0	0	0	1	**7**	1	0	0	1
	C7	0	0	0	0	1	0	**8**	0	0	1
	C8	1	1	0	0	0	0	1	**6**	1	0
	C9	1	0	1	0	0	0	0	0	**8**	0
	C10	0	1	0	0	0	0	1	1	0	**7**
		Precision rate					**74.24%**				
		Recall rate					**72%**				
		True negative rate					**53.41%**				

Table 4.9: Confusion matrix of fuzzy n-Gram for traditional motifs showing number of melodies classified against different classes for Dataset-2.

		Predicted Class									
		C1	C2	C3	C4	C5	C6	C7	C8	C9	C10
Actual Class	C1	**7**	1	0	0	1	0	0	0	0	1
	C2	1	**7**	0	1	1	0	0	0	0	0
	C3	0	1	**8**	0	0	1	0	0	0	0
	C4	0	1	0	**7**	0	0	0	1	1	0
	C5	0	0	0	0	**8**	2	0	0	0	0
	C6	1	0	0	0	0	**9**	0	0	0	0
	C7	0	1	1	0	0	0	**8**	0	0	0
	C8	0	0	0	2	1	0	0	**7**	0	0
	C9	0	0	0	0	1	0	0	1	**7**	1
	C10	1	0	0	0	0	0	0	0	0	**9**
		Precision rate					**78.12%**				
		Recall rate					**77%**				
		True negative rate					**53.61%**				

which is 77% for fuzzy n-Gram as compared to 72% and 50% for weighted and exact n-Gram techniques respectively. However, judging the performance of a technique only through recall rate can often be misleading.

Hence, in order to provide a more balanced interpretation of the confusion matrices F-scores have been calculated and shown in Table 4.10 for Exact n-Gram, Weighted n-Gram and Fuzzy n-Gram. Significantly higher values of F-score for weighted and fuzzy n-Gram

Table 4.10: F-score for different datasets

	Exact n-Gram	Weighted n-Gram	Fuzzy n-Gram
Dataset-1	48.53	62.63	74.36
Dataset-2	42.47	51.58	71.20
Dataset-2 with Predefined catch phrase	50.28	73.10	77.55

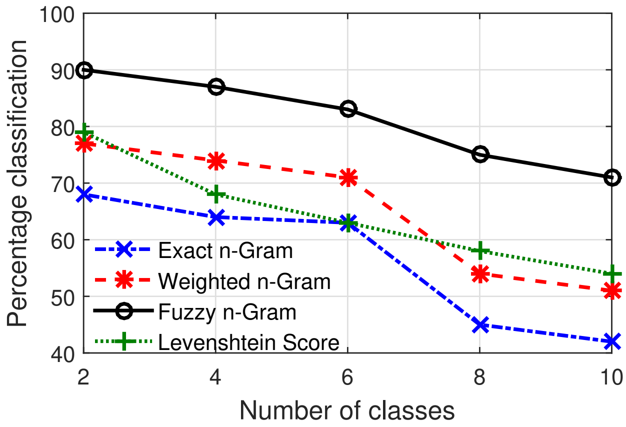

Figure 4.9: Percentage classification results v/s number of classes for Dataset-2.

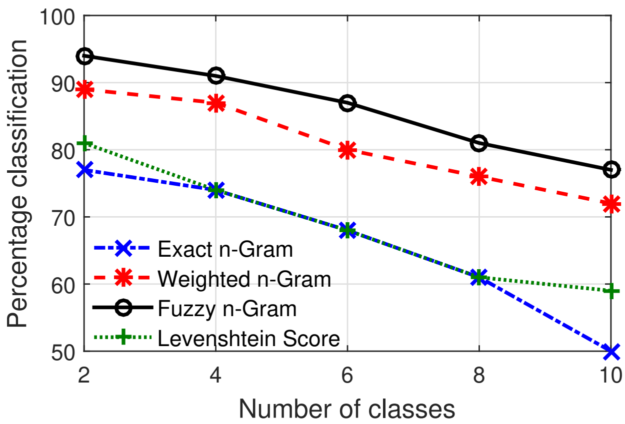

Figure 4.10: Percentage classification results v/s number of classes for Dataset-2 with traditional motifs.

proves the efficacy of entropy based weight assignment and Fuzzification as crucial stages of the proposed technique. The proposed technique performs best in terms of F-score and percentage classification for Dataset-2 with traditional motifs. However, the largest improvement compared to non-fuzzy scheme has been obtained for Dataset-2 with motifs extracted using SIARCT.

In order to visualize the effect of change in total number of classes, we conducted sev-

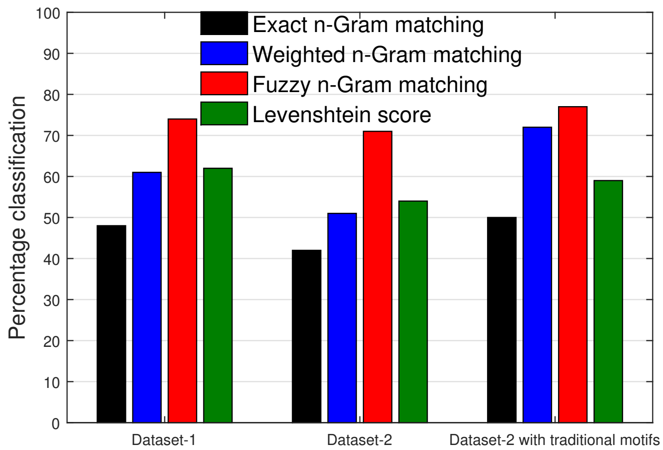

Figure 4.11: Bar graph showing percentage classification for different datasets.

eral experiments by changing the total number of classes and calculating the percentage classification (ratio of total correctly classified melodies to the total number of melodies) for every case. The classes chosen for this experiment were random and any n number of classes were chosen at a time ($n = 2, 4, 6, 8$ *or* 10). Five random draws were done for selecting a particular set of class and any identically recurring set was discarded. The final results are an average of classification success rate over the five randomly drawn sets. This experiment was done to check the accuracy of algorithm for varying number of classes. The maximum number of melodies in a certain class is 10.

For the sake comparison with a popular string matching algorithm we have included results obtained using Levenshtein distance based approximate string matching. Levenshtein distance or edit distance [207] is a well-known measure to compare sequential data. It has been applied in fields such as: automated spell checking [208], DNA analysis and also for music analysis. Application of Levenshtein distance for music includes melodic similarity analysis as described in [209]. Smith *et al.* [197] used levenshtein distance to plain sequences of notes for melodic similarity detection. In 2004 it has been applied to sequences of pitch intervals and contours [124, 210]. Levenshtein distance metric based algorithm as reported in [211] was applied to our database. The algorithm gives better results than exact n-Gram and comparable results with weighted n-Gram which are shown in figure 4.8, 4.9 for Dataset-1, 2 respectively and figure 4.10 for Dataset-2 with traditional motifs/catch phrases.

It can be observed that percentage classification decreases as the number of classes are increased from 2 to 10. This may be due to the reason that pair wise similarity among motifs/catch phrases of different classes increases with an increase in total number of possible classes in a given dataset. The percentage classification results v/s number of classes for Dataset-2 with traditional motifs is shown in Fig. 4.10 where similar trend is observed. Bar graph showing percentage classification for different datasets containing 10 classes with 10 songs in each class is shown in Fig. 4.11 where the results for exact n-Gram, weighted n-Gram and fuzzified n-Gram are presented. It is observed that obtained percentage classification is consistently higher for fuzzified n-Gram as compared to that obtained with exact and weighted n-Gram and Levenshtein distance based algorithm.

Now we summarise the research findings from this paper based on the results. The consistently superior performance of the proposed technique can be attributed to combination of entropy based weight assignment and fuzzification. In entropy based weight assignment we are probabilistically quantifying "importance of match" and then deriving qualitative inferences from those quantified values at different hierarchies. As the final step, transformation of the solution into a high dimensional fuzzy pattern space gives a true estimate of its distance from the ideal solution. Since this technique makes use of variable length pattern matching at several levels of hierarchy any inconsistency in scores manifests itself in the form of some other fuzzy attribute becoming prominent which alters the Euclidean distance of the actual solution from the ideal one.

From the obtained results a couple of observations are worth mentioning:

- Classification results for Indian ragas are better when we use traditional motifs instead of extracted ones.
- Improvement in classification accuracy due to weighting and fuzzification is more when we use motifs extracted using SIARCT.

For Indian ragas, in particular, there is a strong need to identify generic improvisations from characteristic patterns (motifs). This would result in the discovery of more relevant patterns leading to better classification. The concept of an "ideal solution in the pattern space" can be extended to the pattern discovery stage in order to identify a particular fragment of the rendition as either generic modulation or characteristic feature.

Although the present work establishes the efficacy of the probabilistic weighting scheme and subsequent fuzzification, there is scope for further improvement in the pre-processing block.

4.4 Conclusion

n-Gram based template matching is a popular technique used extensively in computational linguistics and plagiarism detection software. The present work essentially validates the efficacy of n-Gram matching in computational musicology. However, the inherent fuzziness in the definition of melodic structures necessitates improvisations in the traditional n-Gram matching. It can be concluded that the calculation of weighted matching scores and construction of the proposed FIS have served as successive stepping stones towards better classification. In our opinion, the proposed technique is not limited by the non-availability of a template for a melodic structure, since we can construct catch phrases based upon the frequency of individual notes for the structures which do not possess a catch phrase traditionally. Therefore, apart from Ragas, the same technique can be applied in the future to other melodic structures available around the world.

Chapter 5

Classification of music structures by using statistical features and distribution estimates

Motifs specific to a Raga or a composer are most of the time more subtle and ephemeral than those belonging to a genre. The author attempted unsupervised identification of Raga pieces using unsupervised techniques without much of a success. However, the same unsupervised models worked for identification of different genres of western music. Therefore in the present chapter, the author reports western music genre classification. The contents of this chapter may provide a background for future workers attempting unsupervised classification of Ragas (any other frameworks involving subtle musical motifs). Classification of music signals based on its genre is a topic studied in great detail over the past few years [24, 31, 39, 40, 157, 212]. Several algorithms and approaches have been implemented for the task of successfully assigning the particular genre for a musical piece. Soft computation algorithms [213], statistical inference [31], machine learning [77], and information theory-based approaches [214] are utilized for genre classification. Pattern matching is one of the popular ways to classify musical pieces. However, the success of the pattern matching stage can only be ensured if the class-specific motifs at our disposal are true representatives of the underlying class. Also, maximum information about the class should be extracted through pattern matching by appropriately quantifying inexact matches [215].

On the other hand, some researchers have used Probability Density Functions (PDF) estimation based classifiers for the music genre. The precise estimation of PDF of a signal is an important step in a lot of signal processing algorithms [216]. The PDF estimation techniques and their resulting models have varying success rates but most of them make the assumption that the PDF to be estimated from samples are Independent and Identically Distributed (IID) [211]. In this Chapter, the author has extracted several temporal, spectral and statistical features from the music dataset. These features were subsequently used to train two classifiers viz. ANN and Gaussian Mixture Model (GMM). These set of features were evaluated with both supervised and unsupervised training paradigms. The GMM classifier was trained to estimate the PDF of observed data. More specifically, a probability distribution estimation based algorithm is developed for genre classification. A Gaussian Mixture Model (GMM) based classifier is used for this task. Instead of directly using original raw data, statistical features like skewness and kurtosis are calculated from data. These features are calculated in time, FFT and Mel Frequency Cepstrum Coefficients (MFCC) domains. Finally, the GMM classifier makes use of these extracted features for classification task. The results calculated from these methods are compared with original raw data and other state of art published literature and are found satisfactory.

5.1 Data acquisition and classification methodology

Music piece is given as input to the preprocessing module in which each music piece is clipped to a length of 30 seconds at 44.1 kHz sampling rate. Further, feature extraction is applied on it. This step includes calculation of Kurtosis and skewness after which FFT and MFCC are calculated. These features are used to extract class specific attributes from the musical piece. The performance of proposed classification systems is evaluated on four different datasets named ballroom dataset [160], SLAC dataset [161], Codaich dataset [162], and Bodhidharma dataset [163]. A general purpose block diagram of the classification methodology has been given in Fig. 5.1. In this chapter the classifier block of Fig. 5.1 will be implemented using two methodologies.

In the next three sections, various features used for analysis in this chapter are described. The features have been broadly grouped into three categories viz. statistical, spectral and temporal feature. All three type of features contain different types of information. Spectral features are more likely to reveal information arising out of musical source (eg. vocal track) whereas statistical feature can be used to distinguish distribution of one musical piece from that of another [217].

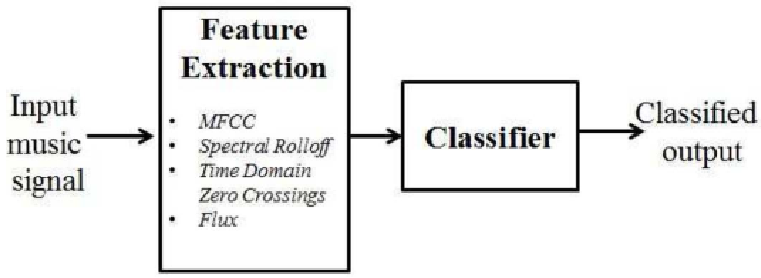

Figure 5.1: Block diagram of the classification system.

5.1.1 Statistical Features

Kurtosis:

Kurtosis is a measure of the combined weight of a distribution's tails relative to the rest of the distribution. Kurtosis is a measure that describes the shape of a distribution's tails in relation to its overall shape. Mathematically, kurtosis for a random variable X is the fourth moment, defined as

$$Kurtosis[X] = \frac{E\left[(X-\mu)^4\right]}{E\left[(X-\mu)^2\right]^2} \tag{5.1}$$

where μ is the mean value of X and $E[\cdot]$ represents the expectation operator.

Skewness:

On the other hand, Skewness is a term in statistics used to describes asymmetry from the normal distribution in a set of statistical data. Mathematically, skewness is defined as

$$Skewness[X] = \frac{E\left[(X-\mu)^3\right]}{E\left[(X-\mu)^2\right]^3/2} \tag{5.2}$$

We used MATLAB for extracting the features Kurtosis, skewness, FFT Coefficients and MFCC [166]. The next step includes probability distribution estimation. For this task, we have taken 5 different distributions namely Gaussian, Cauchy, Gamma, exponential distribution and chi distribution as null hypotheses. Thereafter, Kolmogorov-Smirnov Test is applied on the datasets containing melodies. The Kolmogorov-Smirnov test developed by Chakravart et al. [218] is used to decide if a sample comes from a population with a specific distribution. This test is based on the empirical distribution function (ECDF) which is defined as

$$ECDF = \frac{n(i)}{N} \tag{5.3}$$

where, it is assumed that there are N ordered data points $Y_1, Y_2, \cdots, Y_N$ and $n(i)$ is the number of points less than Y_i and Y_i are ordered from smallest to largest value. One major advantage of this test is that the distribution of the Kolmogorov-Smirnov test statistic does not depend on the CDF being tested. Also, it is an exact test but can only be applied to continuous distributions and tends to be more sensitive near center of distribution.

5.1.2 Spectral Features

Mel-frequency Cepstral coefficients (MFCCs):

MFCCs are considered as a set of dominant feature in speech recognition and are mostly used in music signal processing. The flow chart for calculating MFCCs is shown in Fig.

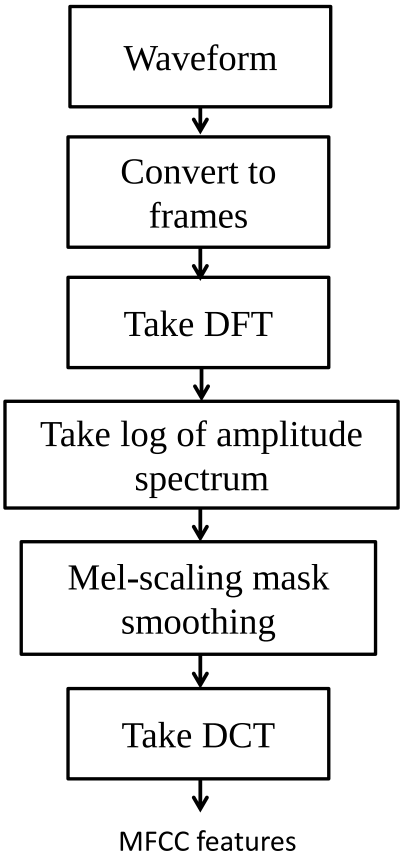

Figure 5.2: The flow chart of calculating MFCCs.

5.2. The features based on short term spectrum are captured by MFCC. MFCC are extensively used in past for the purpose of music and speech analysis. MFCC are generated by the result of decorrelating the mel spectral vectors. Discrete cosine transform (DCT) technique is used for this purpose. Based on short time Fourier transform for each frame the logarithm of amplitude spectrum is taken, according to mel-frequency scaling the frequency bins are grouped and smoothed as given by the relation:

$$f_{mel} = 2595log_{10}\left(1 + \frac{f}{700}\right) \tag{5.4}$$

The inversion from mel scale to cent can be calculated as

$$f_{cent} = 700\left(exp\left(\frac{m}{1125} - 1\right)\right) \tag{5.5}$$

Spectral Roll-off:

It is the measures which represents the spectral shape of a signal. Mathematically, it is

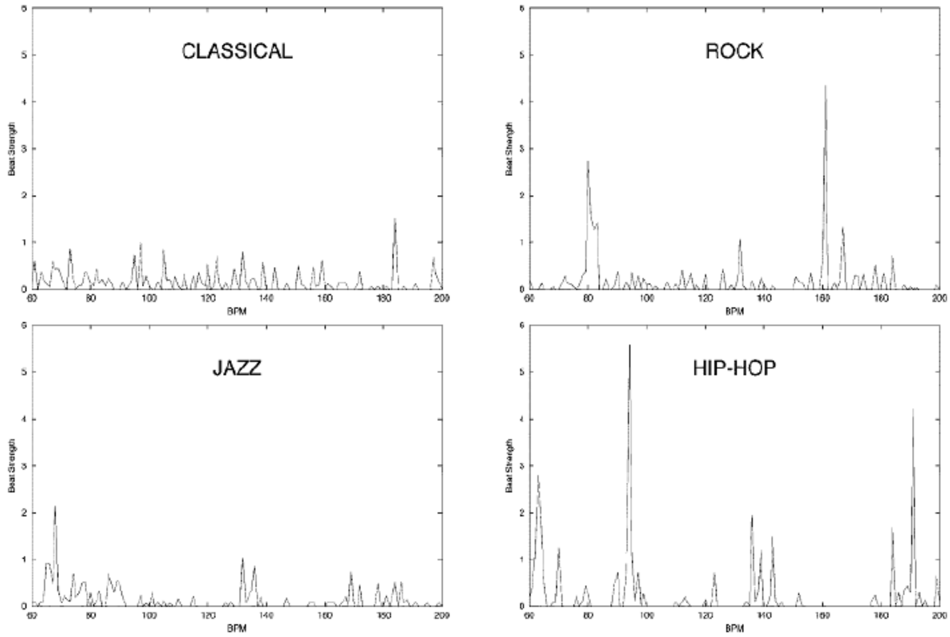

Figure 5.3: Beat histogram of different genres, the horizontal axis is beat per minute (BPM), and the vertical axis is the beat strength [39].

calculated by taking the frequencies below 85% of magnitude as

$$\sum_{n=1}^{R_t} S_t[n] = 0.85 \sum_{n=1}^{M} S_t[n] \tag{5.6}$$

where $S_t[n]$ is defined as the magnitude of the Fourier transform at frequency bin n and frame t.

Beat Histogram:

Beat Histogram is the histogram showing the strength of different rhythmic periodicities in a signal. This is calculated by taking the RMS of 256 windows and then taking the FFT of the result. It is an aggregated histogram of time distances between successive feature local maxima. The histograms maximum position is used to estimate the BPM rate. Fig. 5.3 shows the Beat histogram of different genres, the horizontal axis is beat per minute (BPM), and the vertical axis is the beat strength.

Flux:

Flux is often used to represent the spectral rate of change in a musical piece. Mathematically, it is computed by the formula

$$F_j = \frac{1}{M-1} \sqrt{\sum_{i=1}^{M/2} [\mid X_j(i) \mid - \mid X_{j-1}(i) \mid]^2} \tag{5.7}$$

Bispectrum:
The bispectrum is a useful tool for identifying a process that is either non-Gaussian or is generated by nonlinear mechanisms. Bispectrum, is also known as third-order spectrum, has been shown to have the ability to detect second order nonlinear phase coupling information of the nonlinear system [95].The bispectrum is defined as

$$B(f1, f2) = E\left[X(f1)X(f1)X^*(f1+f2)\right] \tag{5.8}$$

Where $X(f)$ is the Fourier transform of the random signal $x(nT)$, $E[\cdot]$ denotes the expectation operation, and $*$ represents complex conjugate. The bispectrum is a complex-valued function of two frequencies $(f1, f2)$. The frequency is normalized by the Nyquist frequency to be between 0 and 1. If higher order cumulants is absolutely summable

$$\sum_{\tau=-\infty}^{\infty} \cdots \sum_{\tau_{k-1}=-\infty}^{\infty} \left| C_{kx}\left(\tau_1, \cdots, \tau_{k-1}\right) \right| < \infty \tag{5.9}$$

The k-order cumulants spectrum can be defined as $(k-1)$-D Fourier transform of $k$order cumulants as

$$S_{kx}(\omega_1, \cdots, \omega_{k-1}) = \sum_{\tau_1=-\infty}^{\infty} \cdots \sum_{\tau_{k-1}=-\infty}^{\infty} C_{kx}\left(\tau_1, \cdots, \tau_{k-1}\right) \exp\left(-j\sum_{i=1}^{k-1} \omega_i \tau_i\right) \tag{5.10}$$

The third-order spectrum (also known as the bispectrum) can be expressed as

$$B_x(\omega_1, \omega_2) = \sum_{\tau_1=-\infty}^{\infty} \sum_{\tau_2=-\infty}^{\infty} C_{3x}\left(\tau_1, \tau_2\right) \exp\left(-j(\omega_1\tau_1 + \omega_2\tau_2)\right) \tag{5.11}$$

5.1.3 Temporal Feature

Time Domain Zero Crossings:
Time domain zero crossing represents the noisiness of the signal. Is is calculated by using the sign function 0 for negative arguments while a positive argument is given for 1 in the signal. Let us take a signal $x[n]$ in time domain. The time domain zero crossings is calculated for the frame t as

$$TDZC_t = \frac{1}{2}\sum_{n=1}^{M} \mid sign[x[n]] - sign[x[n-1]] \mid \tag{5.12}$$

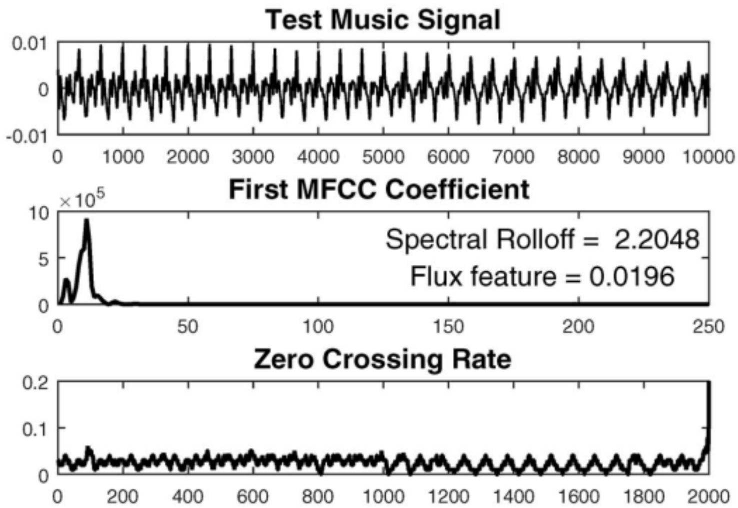

Figure 5.4: Test musical piece 1 and its extracted features.

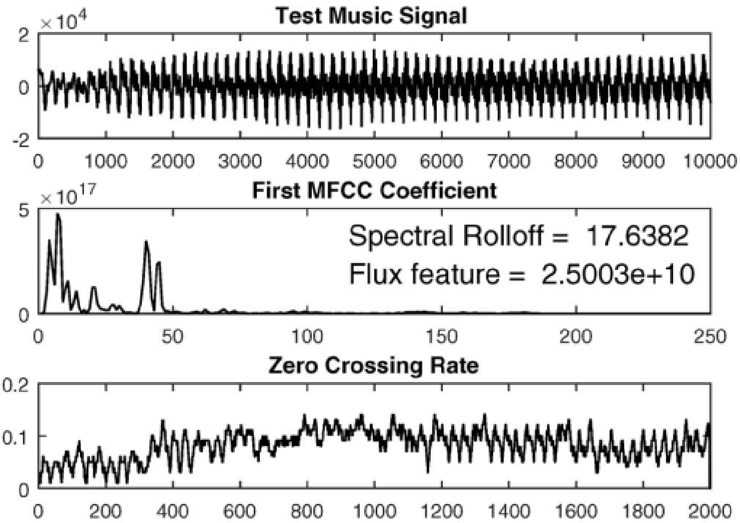

Figure 5.5: Test musical piece 2 and its extracted features.

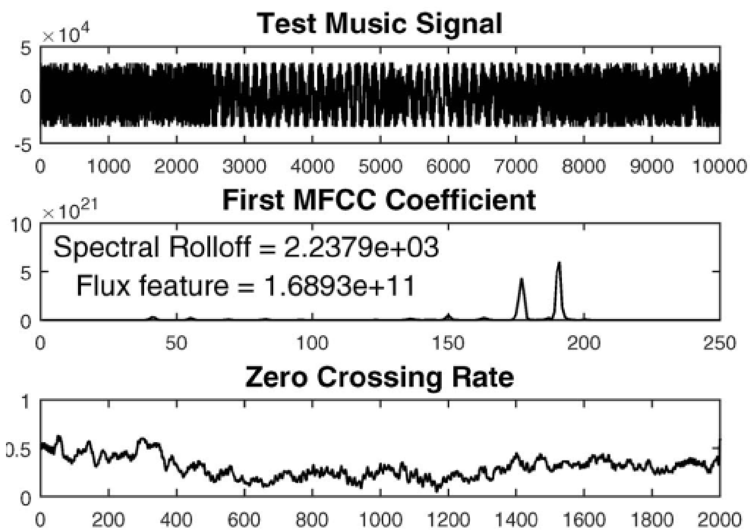

Figure 5.6: Test musical piece 3 and its extracted features.

The extracted features from three random test signals from database are shown in Fig. 5.4, 5.5 and 5.6. Fig. 5.4(a), 5.5(a) and 5.6(a) shows the test musical piece and their corresponding MFCC data is shown in Fig. 5.4(b), 5.5(b) and 5.6(b). Time Domain Zero Crossings are shown in Fig. 5.4(c), 5.5(c) and 5.6(c). Rest of the features like spectral rolloff, flux features are shown in the legend of Fig. 5.4(b), 5.5(b) and 5.6(b).

5.2 Probability Density Function Estimation

Probability density function (PDF)estimation can be used as an effective precursor to classification. The estimation generally involves two phases, the learning of a distribution and subsequently inference of probabilities of certain configurations within the learned model. According to Vapnik [219], it is certainly better to learn the quantity you are interested in rather than go sample by sample through a harder problem. This section describes PDF estimation based approach for drawing inferences about the genre of the test samples. Three distributions viz. Normal, Cauchy, and Generalized Gamma were chosen as null hypotheses.

This test checks whether a set of observations belongs to a continuous distribution or not. This test can be applied on data sets with small size and proves to be a more powerful test as compared to chi-square test [220]. It gives the cumulative distribution function $Pr(K \leq x)$ of the hypothesized distribution and the empirical distribution function $F_{data}(x)$ of the observed data. For n iid random variables, the distribution of the random variables is given as [221]

$$F_n(x) = \frac{1}{n}\sum_{i=1}^{n} I(X_i) \tag{5.13}$$

where $I(X_i)$ is the indicator function which is equal to 1 if $X_i \leq x$ else 0. Let us assume the variables to be the Kolmogorov distribution. The CDF is given by [220]

$$Pr(K \leq x) = 1 - 2\sum_{k=1}^{\infty}(-1)^{k-1}e^{-2k^2x^2} \tag{5.14}$$

The goodness-of-fit test can be constructed by using the critical values of the Kolmogorov distribution. This test is asymptotically valid when $n \to \infty$. It rejects the null hypothesis at level α if $\sqrt{n}D_n > K_\alpha$ where K_α is found from $Pr(K \leq K_\alpha) = 1 - \alpha$ [221]. The steps of probability density function estimation are shown using a flow chart in Fig. 5.7. The histogram of five different songs from ten different genres is shown in Fig. 5.8 to 5.17. Data on x-axis of the Fig. 5.8 to 5.17 is the instantaneous normalized amplitude values of a musical file chosen from a particular class whereas y-axis shows the

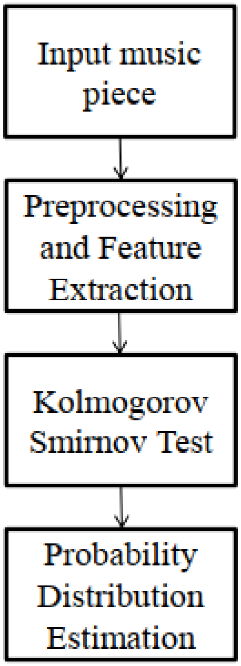

Figure 5.7: Steps of probability density function estimation.

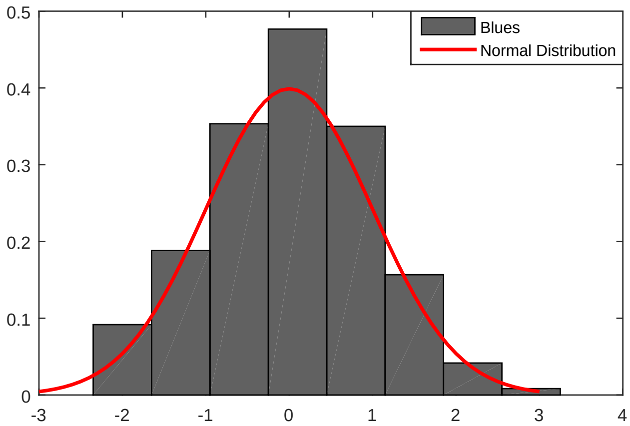

Figure 5.8: Histogram of Blues.

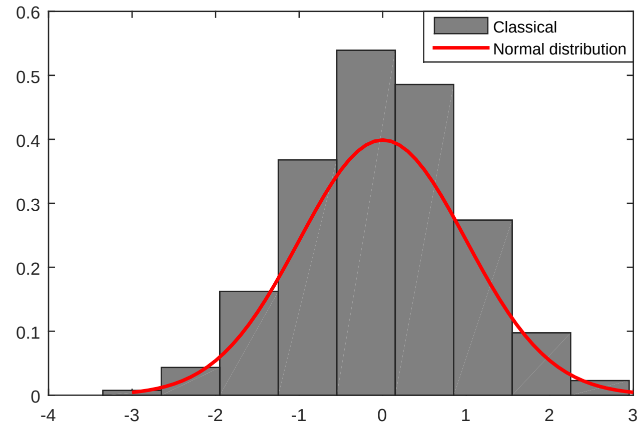

Figure 5.9: Histogram of Classical.

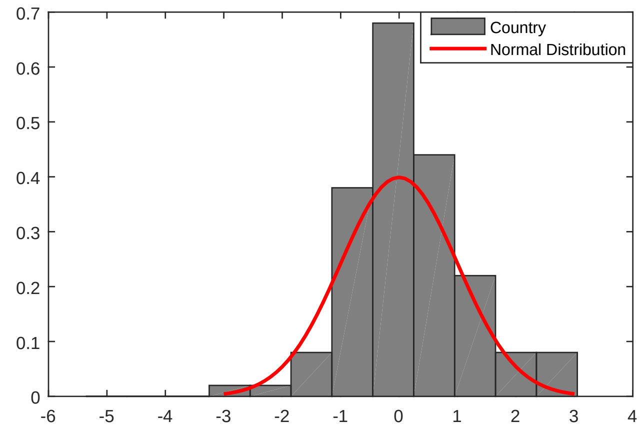

Figure 5.10: Histogram of Country.

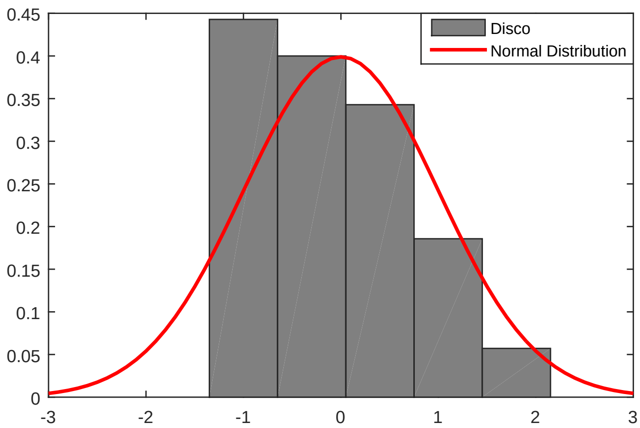

Figure 5.11: Histogram of Disco.

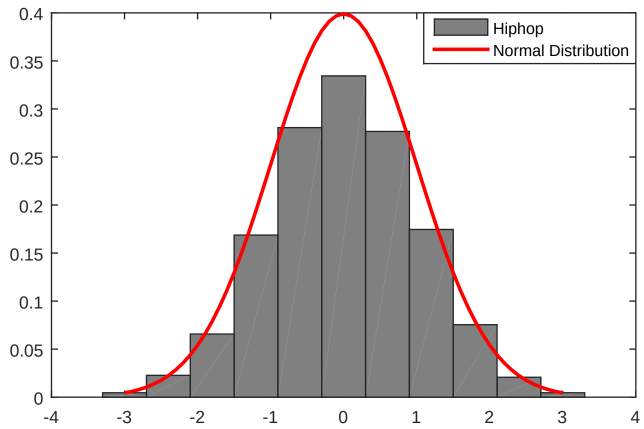

Figure 5.12: Histogram of Hiphop.

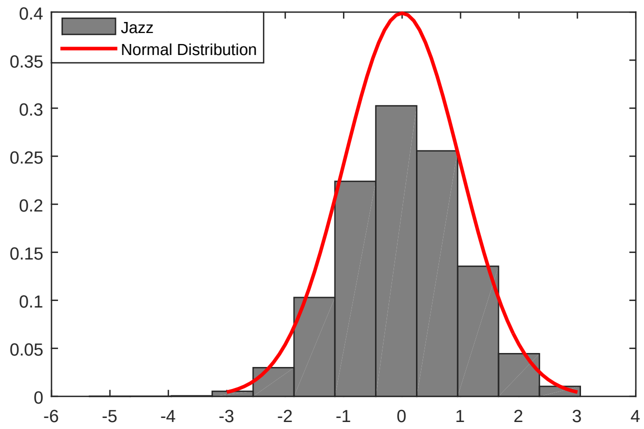

Figure 5.13: Histogram of Jazz.

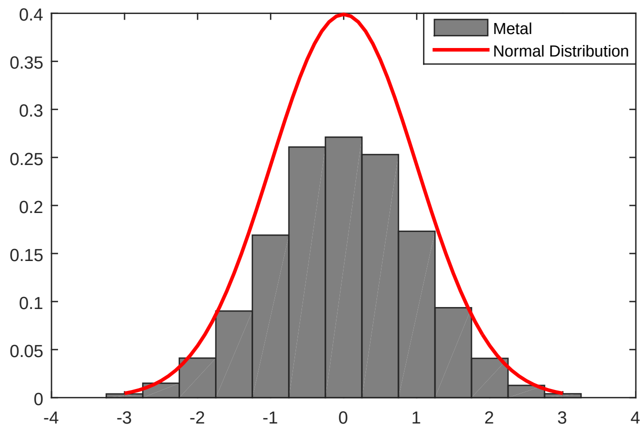

Figure 5.14: Histogram of Metal.

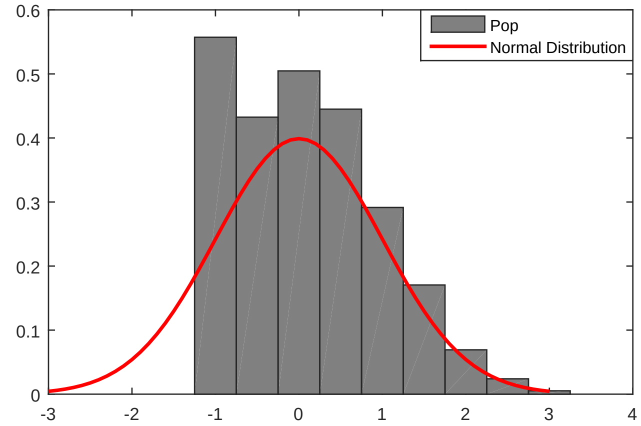

Figure 5.15: Histogram of Pop.

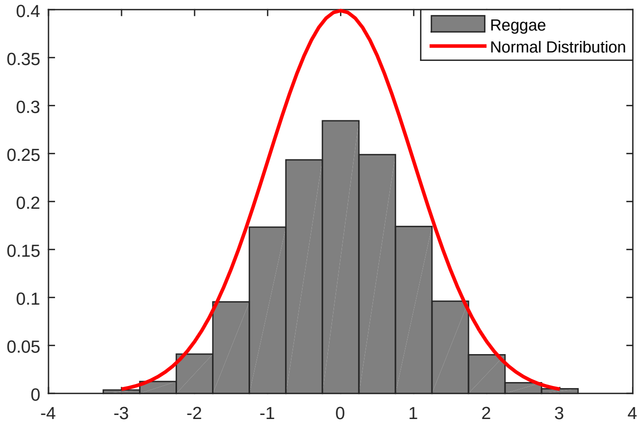

Figure 5.16: Histogram of Reggae.

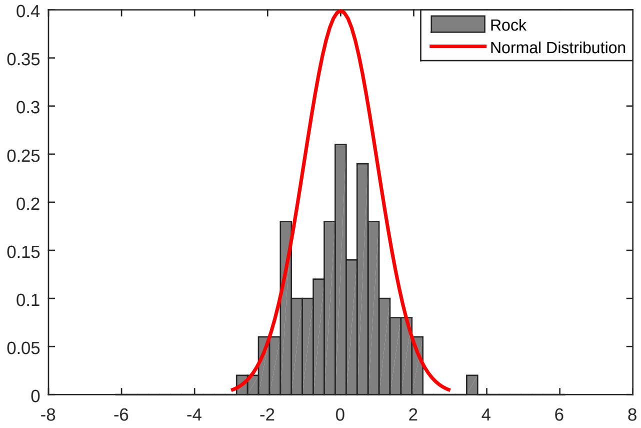

Figure 5.17: Histogram of Rock.

corresponding density function.

It can be visualized from Fig. 5.8 to 5.17 that the shape of histogram for each musical file is different. This difference in shape of histogram can be further seen when compared with the Gaussian PDF in each plot. This gives an inspiration for using PDF of a musical file as a unique feature to classify them based on genre.

5.3 The Classifiers

After obtaining the above mentioned features, the classification was done employing two popular classifiers viz. Gaussian Mixture Model (GMM) and Artificial Neural Network (ANN). The purpose of choosing these classifiers was to compare the performance of one supervised paradigm (ANN) with an unsupervised one (GMM). Our ultimate goal was to made unsupervised classification better the supervised one. This would ensure suitability of the reported classifier for real time deployment.

5.3.1 GMM

A lot of classifiers are used in literature for the task of music genre classification. These include Linear Discriminant Analysis (LDA) [222], Gaussian Mixture Models (GMM) [223], Support vector machines (SVMs) [61] and K Nearest Neighbor (KNN) [224]. Out of these classifiers, GMM has gained a lot of attention specially in the task of genre classification.

The pdf of each genre class is assumed to be multidimensional gaussian distribution in GMM. The parameters of these distributions are calculated by using the training data set [82]. In order to estimate the parameters for each Gaussian component and the mixture weight, an iterative expectation maximization (EM) algorithm is used. Musical piece is given as an input to the genre classification system. Initially, features from the input musical piece are extracted. Based on these features, the classifier (GMM) takes a decision of the class to which the given input music piece belongs to.

In order to define a GMM, K Gaussians are superimposed together as:

$$p(x) = \sum_{k=1}^{K} \pi_k \mathcal{N}\left(x|\mu_k, \Sigma_k\right) \tag{5.15}$$

Where, $\mathcal{N}$ is a given Gaussian with mean μ_k and variance $\sum_k$. The aggregation of these Gaussians is done by using mixing weights given as $\pi_k \in [0,1]$ and $\sum k = 1K\pi_k = 1$. The values of μ_k, Σ_k and π_k are initialized by the EM algorithm. After replacing old parameters by new ones we calculate log-likelihood

$$lnP(X_1, X_2, \cdots, X_n) = \sum_{n=1}^{N} ln \sum_{k=1}^{K} \pi_k \mathcal{N}(x_n|\mu_k, \Sigma_n) \tag{5.16}$$

and check for stopping criteria.

Inspired by the work of Glodek et al. [225], the author has also employed the same technique to obtain the model for PDF estimation of individual features of musical piece. Since the observation data consists of different statistical, temporal and spectral features of the musical piece, a single GMM model for genre classification is more likely to get stuck in local minima. In this scenario, the use of an ensemble of individual mixture models may create a more stable and accurate final model. Previous work [225] has reported better performance of the ensemble model in estimating non-Gaussian distributions than a single GMM.

The GMM ensembling is done through the following steps:

1. Let $\mathcal{L}$ be the number of potential ensemble members.
2. Generate initial $\mathcal{L}$ GMMMs, $g_1, \cdots, g_{\mathcal{L}}$
3. Compute initial means and covariance for each g_i
4. Compute all GMM $g_1, \cdots, g_{\mathcal{L}}$ with maximal log-likelyhood.
5. Compute weights $w_1, \cdots, w_m$ by using $w_i = \frac{1}{M}$
6. Get GMM ensemble $g = \sum_{i=1}^{M} w_i g_i$.

5.3.2 ANN

Artificial neural networks have known to be very successful in the field of pattern recognition [57]. The success of ANN for musical data classification is also presented and verified in Chapter 3. ANN can be trained to classify the new inputs not used for training [36]. Features extracted from musical piece can be used to train an ANN in order to establish a genre based classifier [185]. In order to create a genre classifier system based on ANN, following steps are used [190]:

- Create a dataset of musical pieces (explained in Section 1.14)
- Normalize the data
  In normalization, the values in dataset are set to lie in the range from 0 to 1 by

using the relation:

$$X_n = \frac{X - X_{min}}{X_{max} - X_{min}} \tag{5.17}$$

Where X is the value (musical piece) that should be normalized, X_n is the normalized value, X_{min} and X_{max} are the minimum and maximum values of X respectively.

- Extract features from the musical pieces of dataset. For this purpose, FFT and MFCC are applied on each musical piece of the dataset.

- Divide the dataset into training and testing datasets. For this purpose, the complete dataset was partitioned randomly into subsets, i.e. training set consist of roughly 80% samples, and 20% for validation and test sets each. Further fivefold cross-validation is employed in these datasets to remove the problem of overfitting.

- Create a neural network
  MATLAB provides a very user friendly environment to implement an ANN. There are three layers in an ANN: input layer, hidden layer and output layer. The configuration of the input layer is dependent on the shape of the training data. The number of neurons in input layer is same as number of dimensions in the dataset. The number of neurons in output layer is chosen equal to the number of genre classes which are present in the dataset. There are a lot of rules to choose number of hidden layers and the number of neurons in each hidden layer. For the task of genre classification in this dissertation, one hidden layer is taken and the number of neurons in the hidden layer is taken as the mean of neurons in input and output layers.

- Train the ANN

We have trained the network 20 times each with different architecture. The architecture was changed by changing the number of neurons in the hidden layer from two to eight. Then simulate the best performance observed from 20 trained results, to get the percentage classification.

5.4 Simulation Results and Discussion

Table 5.1-5.4 shows the average kurtosis and skewness results of different genres for ballroom, SLAC, Codaich and Bodhidharma datasets respectively. An interesting observation from Table 5.1-5.4 can be made that the numerical values of kurtosis and skewness are distinct for almost each genre. Some overlap is observed in time domain analysis but a clear distinction is seen in the case of FFT domain.

Table 5.1: The average kurtosis and skewness results of different genres for ballroom dataset.

	Time domain		FFT		MFCC		K-S Test (Time)	K-S Test (FFT)	K-S Test (MFCC)
	Kurtosis	Skew	Kurtosis	Skew	Kurtosis	Skew			
Blues	12.35	-10.6	27.77	-22.78	21.56	10.61	-4.174	-0.244	-3.3514
Classical	3.05	-12.7	18.47	-29.68	28.16	41.51	-3.898	0.398	-2.7304
Country	21.65	-1.9	7.07	3.62	16.76	-11.59	-3.991	0.056	-3.6064
Disco	6.95	-1	25.37	43.52	17.66	-21.49	-3.955	0.653	-3.8104
Hiphop	23.75	1.1	24.17	53.12	38.96	11.51	-4.006	1.592	-4.4674
Jazz	4.55	-15.4	27.47	-30.58	-14.44	2.81	-2.956	-0.244	-3.3544
Metal	9.65	-10	112.07	51.62	-7.54	13.01	-3.949	0.053	-2.1514
Pop	2.45	-10.9	77.87	-50.98	32.66	-2.59	-4.087	0.689	-3.9094
Reggae	8.15	3.5	149.57	-19.18	67.76	40.61	-3.874	0.446	-3.2524
Rock	3.05	-18.7	67.97	-10.78	28.76	75.41	-3.982	1.031	-2.5594

Table 5.2: The average kurtosis and skewness results of different genres for SLAC dataset.

	Time domain		FFT		MFCC		K-S Test (Time)	K-S Test (FFT)	K-S Test (MFCC)
	Kurtosis	Skew	Kurtosis	Skew	Kurtosis	Skew			
Blues	14.515	-6.14	28.393	-17.102	22.804	12.949	-0.3566	3.1804	0.38374
Classical	6.145	-8.03	20.023	-23.312	28.744	40.759	-0.1082	3.7582	0.94264
Country	22.885	1.69	9.763	6.658	18.484	-7.031	-0.1919	3.4504	0.15424
Disco	9.655	2.5	26.233	42.568	19.294	-15.941	-0.1595	3.9877	-0.02936
Hiphop	24.775	4.39	25.153	51.208	38.464	13.759	-0.2054	4.8328	-0.62066
Jazz	7.495	-10.46	28.123	-24.122	-9.596	5.929	0.7396	3.1804	0.38104
Metal	12.085	-5.6	104.263	49.858	-3.386	15.109	-0.1541	3.4477	1.46374
Pop	5.605	-6.41	73.483	-42.482	32.794	1.069	-0.2783	4.0201	-0.11846
Reggae	10.735	6.55	138.013	-13.862	64.384	39.949	-0.0866	3.8014	0.47284
Rock	6.145	-13.43	64.573	-6.302	29.284	71.269	-0.1838	4.3279	1.09654

Similarly, the values from MFCC based coefficients also show distinct features for each genre. Apart from this, the results based on KolmogorovSmirnov test for all three cases are also given in Table 5.1-5.4. The values of K test signify the similarity of given PDF with normal PDF. These distinct values of K test are used to classify a given melody into the genre class. The data pre-processed by FFT and MFCC on raw data were fed to the artificial neural network (ANN) classifier. The classifier was trained to 500 epochs with training function train LM being used. The number of neurons in the hidden layer was varied experimentally and a particular architecture was trained with input data a number of times.

The ANN classification results of different techniques like raw data, Bispectrum and MCC is shown in Table 5.5 for ballroom, SLAC, Codaich and Bodhidharma datasets. The confusion matrix of ballroom, SLAC, Codaich and Bodhidharma datasets are shown in Table 5.6-5.9. The architecture, at which minimum error training was obtained, was then simulated with the test data and actual classification performance was noted.

The confusion matrices shown in Table 5.6-5.9 gives the information about number of correct predicted and number of wrong predicted class by the classifier for the four datasets which performed best in terms of classification percentage from all the four different

Table 5.3: The average kurtosis and skewness results of different genres for Codaich dataset.

	Time domain		FFT		MFCC		K-S Test (Time)	K-S Test (FFT)	K-S Test (MFCC)
	Kurtosis	Skew	Kurtosis	Skew	Kurtosis	Skew			
Blues	10.945	-9.4556	43.72522	-26.33708	35.11816	19.94146	-0.549164	4.897816	0.5909596
Classical	2.575	-12.3662	30.83542	-35.90048	44.26576	62.76886	-0.166628	5.787628	1.4516656
Country	19.315	2.6026	15.03502	10.25332	28.46536	-10.82774	-0.295526	5.313616	0.2375296
Disco	6.085	3.85	40.39882	65.55472	29.71276	-24.54914	-0.24563	6.141058	-0.0452144
Hiphop	21.205	6.7606	38.73562	78.86032	59.23456	21.18886	-0.316316	7.442512	-0.9558164
Jazz	3.925	-16.1084	43.30942	-37.14788	-14.77784	9.13066	1.138984	4.897816	0.5868016
Metal	8.515	-8.624	160.56502	76.78132	-5.21444	23.26786	-0.237314	5.309458	2.2541596
Pop	2.035	-9.8714	113.16382	-65.42228	50.50276	1.64626	-0.428582	6.190954	-0.1824284
Reggae	7.165	10.087	212.54002	-21.34748	99.15136	61.52146	-0.133364	5.854156	0.7281736
Rock	2.575	-20.6822	99.44242	-9.70508	45.09736	109.75426	-0.283052	6.664966	1.6886716

Table 5.4: The average kurtosis and skewness results of different genres for Bodhidharma dataset.

	Time domain		FFT		MFCC		K-S Test (Time)	K-S Test (FFT)	K-S Test (MFCC)
	Kurtosis	Skew	Kurtosis	Skew	Kurtosis	Skew			
Blues	2.73625	-4.822356	13.117566	-0.7901124	1.0535448	0.5982438	-0.01647492	0.14693448	0.017728788
Classical	0.64375	-6.306762	9.250626	-1.0770144	1.3279728	1.8830658	-0.00499884	0.17362884	0.043549968
Country	4.82875	1.327326	4.510506	0.3075996	0.8539608	-0.3248322	-0.00886578	0.15940848	0.007125888
Disco	1.52125	1.9635	12.119646	1.9666416	0.8913828	-0.7364742	-0.0073689	0.18423174	-0.001356432
Hiphop	5.30125	3.447906	11.620686	2.3658096	1.7770368	0.6356658	-0.00948948	0.22327536	-0.028674492
Jazz	0.98125	-8.215284	12.992826	-1.1144364	-0.4433352	0.2739198	0.03416952	0.14693448	0.017604048
Metal	2.12875	-4.39824	48.169506	2.3034396	-0.1564332	0.6980358	-0.00711942	0.15928374	0.067624788
Pop	0.50875	-5.034414	33.949146	-1.9626684	1.5150828	0.0493878	-0.01285746	0.18572862	-0.005472852
Reggae	1.79125	5.14437	63.762006	-0.6404244	2.9745408	1.8456438	-0.00400092	0.17562468	0.021845208
Rock	0.64375	-10.547922	29.832726	-0.2911524	1.3529208	3.2926278	-0.00849156	0.19994898	0.050660148

Table 5.5: Percentage classification of ANN results of different techniques.

		Ballroom dataset			SLAC dataset			Codaich dataset			Bodhidharma dataset		
S.No	Neurons	Rawdata	Bispectrum	MCC	Rawdata	Bispectrum	MCC	Rawdata	Bispectrum	MCC	Rawdata	Bispectrum	MCC
1	2	51	32	63	59	37	67	56	30	47	31	61	53
2	3	42	37	58	54	62	62	43	55	52	35	64	58
3	4	34	52	59	55	32	63	47	25	55	32	57	73
4	5	56	51	67	63	70	71	44	63	61	37	68	72
5	6	62	64	69	65	59	73	44	52	64	34	71	75
6	7	57	59	56	52	68	60	56	58	58	41	69	78
7	8	52	66	52	48	47	56	56	40	51	43	65	67

Table 5.6: Confusion matrix of Ballroom dataset

		Predicted Class									
		Blues	Classical	Country	Disco	Hiphop	Jazz	Metal	Pop	Reggae	Rock
Actual Class	Blues	72	4	5	3	1	4	3	2	2	4
	Classical	5	82	1	2	0	3	1	2	2	2
	Country	4	7	71	5	2	4	1	1	2	3
	Disco	2	3	3	77	1	4	3	2	1	4
	Hiphop	2	1	1	2	81	1	2	7	1	2
	Jazz	3	2	0	5	2	79	4	2	2	1
	Metal	1	2	3	4	5	2	74	4	3	2
	Pop	2	3	2	2	4	3	6	69	5	4
	Reggae	2	3	5	3	3	4	3	4	69	4
	Rock	6	4	5	6	2	5	3	2	3	64

Table 5.7: Confusion matrix of SLAC dataset

		Predicted Class									
		Blues	Classical	Country	Disco	Hiphop	Jazz	Metal	Pop	Reggae	Rock
Actual Class	Blues	62	5	2	4	2	6	6	3	4	6
	Classical	1	74	4	7	3	1	5	1	0	4
	Country	1	3	68	2	6	7	3	5	1	4
	Disco	5	1	2	64	4	6	7	3	2	6
	Hiphop	4	3	7	2	69	3	5	3	4	0
	Jazz	2	0	3	2	7	70	5	6	3	2
	Metal	0	3	5	6	2	1	62	6	7	8
	Pop	7	6	3	4	7	6	0	59	2	6
	Reggae	4	1	7	5	4	5	6	0	65	3
	Rock	8	5	2	7	0	8	6	0	7	57

Table 5.8: Confusion matrix of Codaich dataset

		Predicted Class									
		Blues	Classical	Country	Disco	Hiphop	Jazz	Metal	Pop	Reggae	Rock
Actual Class	Blues	58	4	6	5	4	5	7	3	4	4
	Classical	7	70	3	4	0	3	5	0	3	5
	Country	0	3	64	3	1	7	4	7	5	6
	Disco	2	7	6	60	2	4	5	2	4	8
	Hiphop	1	4	0	5	65	7	6	4	3	5
	Jazz	6	5	4	5	2	66	7	2	3	0
	Metal	6	6	4	2	3	3	58	3	6	9
	Pop	6	3	2	7	4	5	7	55	5	6
	Reggae	5	6	4	5	7	0	4	7	61	1
	Rock	7	2	7	0	5	9	4	6	7	53

Table 5.9: Confusion matrix of Bodhidharma dataset

		Predicted Class									
		Blues	Classical	Country	Disco	Hiphop	Jazz	Metal	Pop	Reggae	Rock
Actual Class	Blues	56	8	6	5	5	7	1	3	5	4
	Classical	7	68	3	4	3	5	6	1	3	0
	Country	0	5	62	3	6	5	6	5	3	5
	Disco	4	0	5	58	3	4	5	7	5	9
	Hiphop	1	0	1	4	63	2	6	9	8	6
	Jazz	4	6	5	6	4	64	4	5	2	0
	Metal	3	6	2	0	4	6	56	8	6	9
	Pop	1	4	7	5	6	3	5	53	6	10
	Reggae	0	6	3	4	8	5	6	7	58	3
	Rock	4	5	6	6	7	4	3	6	5	54

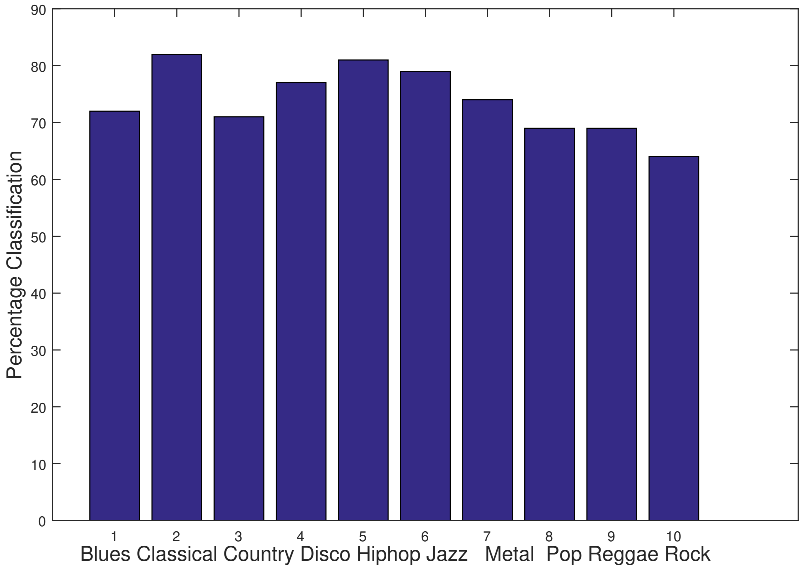

Figure 5.18: Percentage classification of Ballroom dataset using GMM.

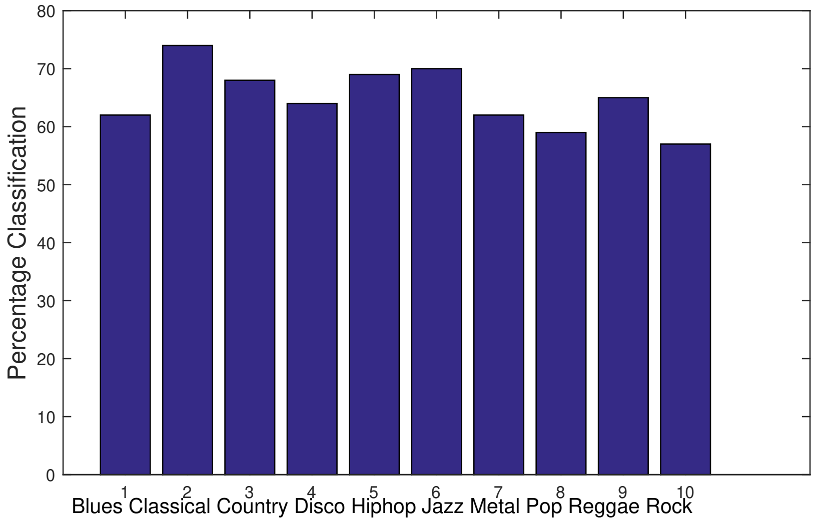

Figure 5.19: Percentage classification of SLAC dataset using GMM.

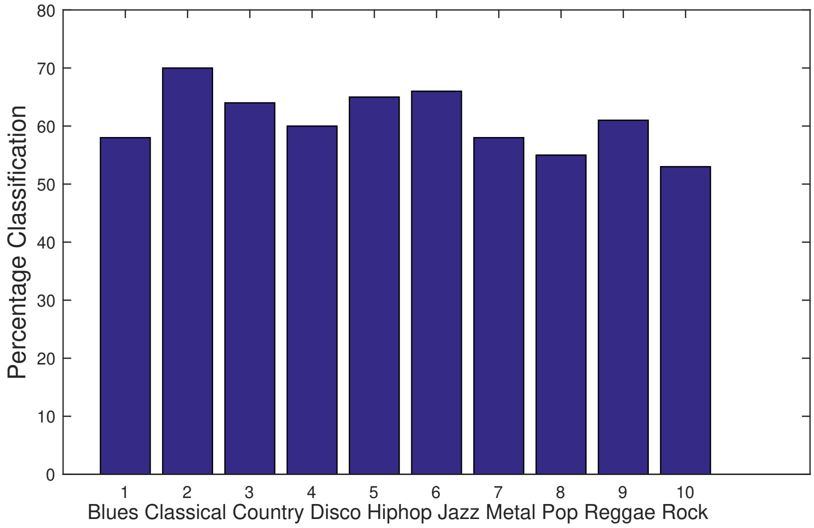

Figure 5.20: Percentage classification of Codaich dataset using GMM.

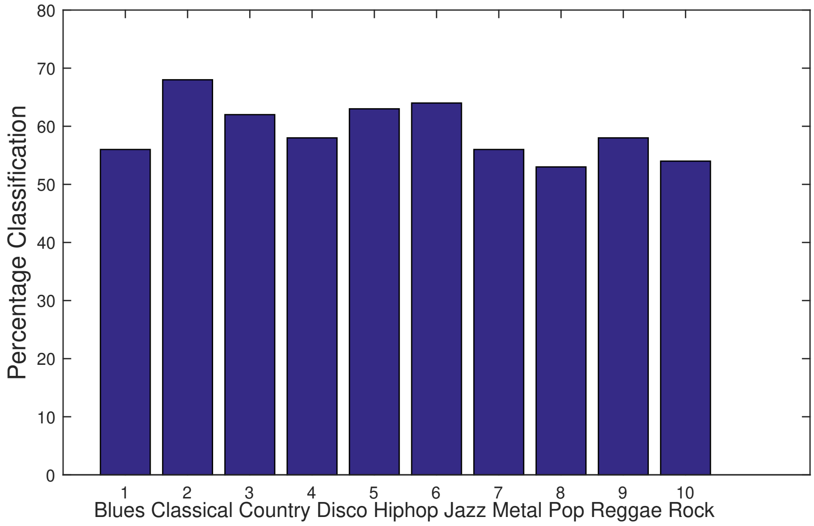

Figure 5.21: Percentage classification of Bodhidharma dataset using GMM.

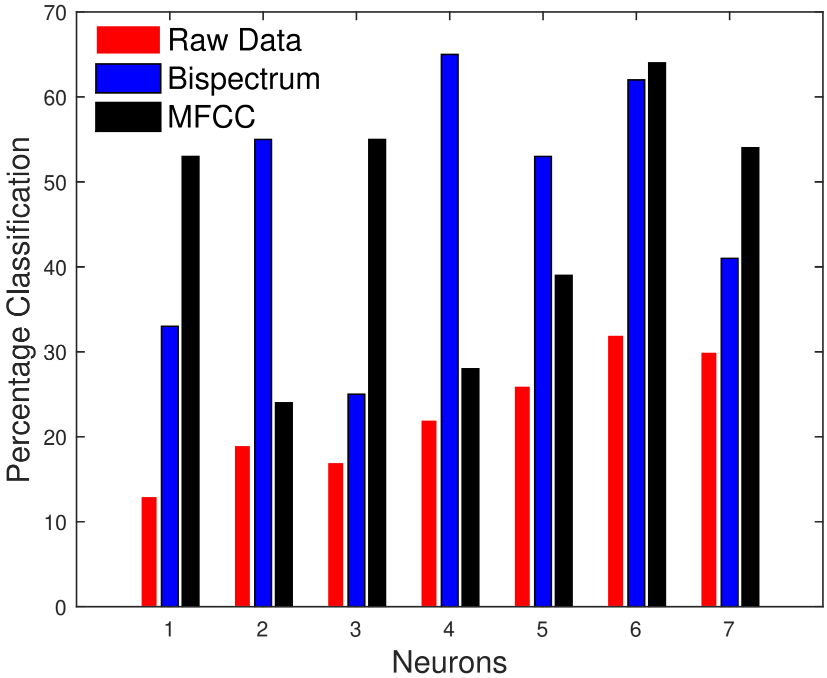

Figure 5.22: Percentage classification results of ANN classifier on Ballroom dataset with statistical features.

datasets. The columns represents the predicted musical piece genre by the classifier and the rows represents the actual musical piece genre predefined by human experts. The diagonal values of confusion matrix represents the correct percentage classification of a particular class.

The bar graph in Fig. 5.18-5.21 show the percentage classification of the four datasets using GMM. The classification rate is found to be between 70-80 % from the GMM based classifier which is superior when compared with the approaches from Jensen [226] giving 48% classification, Holzapfel [227] with 66.48% and Marchand [228] with 75.52% classification on Ballroom dataset. A similar classification percentage as reported in this work is achieved in [229] using 111 features at the cost of increased complexity as compared to our algorithm which uses less number of features. Authors in [133] report a 60-70% classification rate FFT and MFCC on same datasets as used in this work. The bar graph of Fig. 5.22 shows the percentage classification v/s number of neurons for Ballroom dataset when the raw data and data obtained from statistical features i.e. Bispectrum and MFCC were fed to ANN classifier. It can be observed from Fig. 5.22 that the percentage classification of raw data is minimum as it improves for Bispectrum and MFCC. Another important observation is that ANN gives slightly better classification results when compared with GMM for same dataset.

5.5 Conclusion

In this chapter, the GMM has been described for genre classification. It was observed that the performance of classifier can be improved by feeding extracted features from original data. Another important observation is that the system works differently in different domains. Therefore, the classification system is implemented in time, FFT and MFCC domains with the help of statistical features of musical piece. Experimental results obtained from four different standard datasets named ballroom dataset [160], SLAC dataset [161], Codaich dataset [162], and Bodhidharma dataset [163] show promising results. The simplicity and ease of implementation of this classifier (due to less number of features used as compared to existing techniques) makes it a suitable candidate in the construction of online music libraries and maintaining digital music collections based on genre.

Chapter 6

Conclusions and Future Scope

6.1 Conclusions

This thesis embodies efforts towards the automatic classification of music structures across some popular frameworks. Initially motivated by the need of a Raga identification system, the author has developed and applied several classification techniques for genre and composer classification as well. Since Raga is known to have some sort of a representative pattern, most of the efforts of the author were directed towards the development of pattern matching based techniques which are reported in chapters 3 and 4 of the thesis. The author's motivation for developing and improving n-Gram based pattern matching comes from the suboptimal and non-reliable performance of ANN based classifiers which also required a large training set with pre-defined labels. Promising results obtained from the improved n-Gram matching technique underscores the usefulness of a robust pattern-matching paradigm.

No doubt, modeling of the classification problem as an Analytic Hierarchy Process and identification of positive and negative ideal solutions proved to be decisive factors in ensuring the success of the technique. The efficacy of the techniques reported in Chapter 3 and 4 was validated by its good performance in composer classification also where the musical pieces were drawn from western classical music. Fig. 6.1 depicts the results of PCA, DCT, MCC on raw data and MCC on DCT on the Raga dataset and MIDI dataset for ANN classifier. Fig. 6.2 shows the results of composer dataset, raga dataset and raga dataset with the predefined catch phrase for exact, weighted and fuzzy n-Gram for the sake of comparison and analysis by the reader.

Best classification results obtained from each scheme suggest that pattern matching scheme (improved using weighting and fuzzification) works better for ragas. Interestingly, ragas are such frameworks that support a "representative pattern". This finding based on an average of 5 fold cross-validation cannot just be a coincidence.

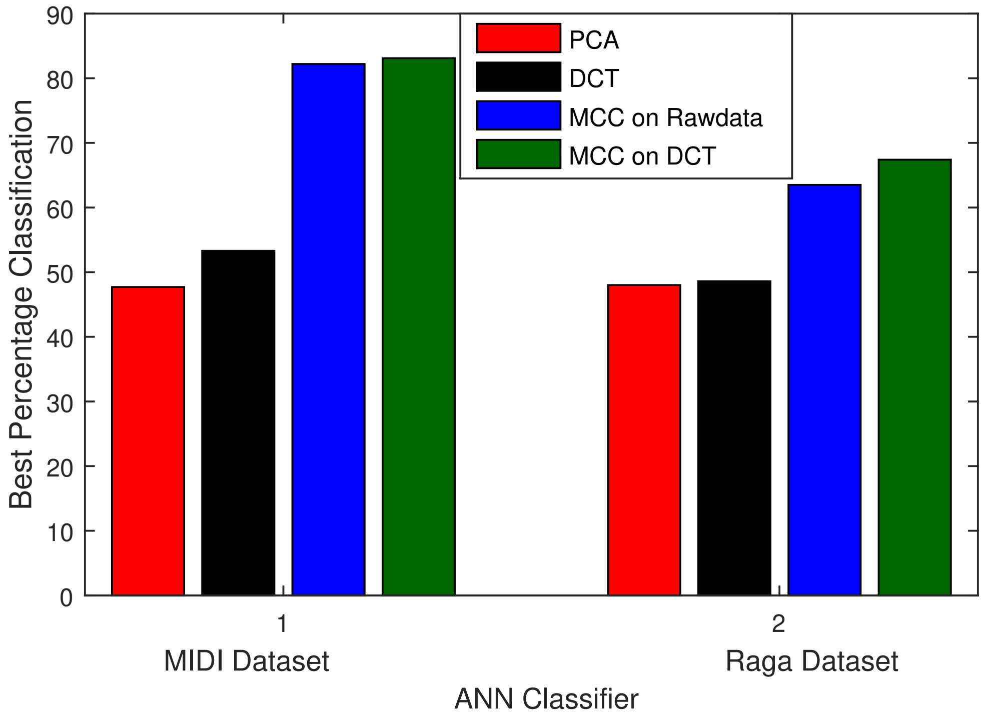

Figure 6.1: Results of PCA, DCT, MCC on raw data and MCC on DCT on Raga dataset and MIDI dataset for ANN classifier.

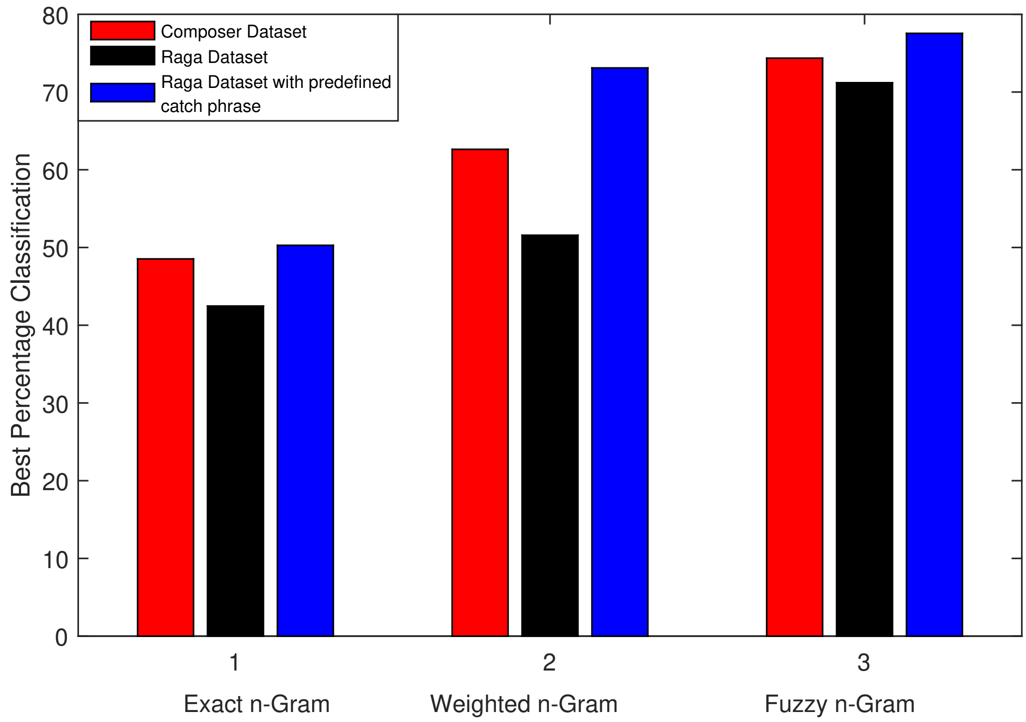

Figure 6.2: Results of composer dataset, raga dataset and raga dataset with the predefined catch phrase for exact, weighted and fuzzy n-Gram.

6.2 Future Scope

Although the machine learning and pattern matching based classifiers seem to work well with a small number of classes (up to 10), the performance falls drastically with an increasing number of classes which calls for more investigation into the techniques reported in this dissertation.

As a future step, the author envisages the development of efficient mixture models for unsupervised classification of different melodies structures. For this purpose, the statistical parameters and features reported in chapter 5 could prove to be useful for the research community. Furthermore, the performance of deep learning can be evaluated for music classification in the future.

In the n-gram method, more sophisticated aggregation techniques and complex classifiers can be utilized to achieve better classification results. In the current study, raga, genre and composers are taken to be as the generic classes for the given melodic datasets. In future studies, other classes based on different attributes of melodies can be chosen.

References

[1] George Tzanetakis and Perry Cook. Musical genre classification of audio signals. *IEEE Transactions on speech and audio processing*, 10(5):293–302, 2002.

[2] J Stephen Downie. Music information retrieval. *Annual review of information science and technology*, 37(1):295–340, 2003.

[3] 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.

[4] Meinard Mller. *Information Retrieval for Music and Motion*. Springer Berlin Heidelberg, 2007.

[5] Toni Heittola. Automatic classification of music signals. *Master of Science Thesis*, 2003.

[6] Kaustuv Kanti Ganguli, Abhinav Rastogi, Vedhas Pandit, Prithvi Kantan, and Preeti Rao. Efficient melodic query based audio search for hindustani vocal compositions. In *ISMIR*, pages 591–597, 2015.

[7] Akhilesh K Sharma and Prakash Ramani. Rigorous data analysis and performance evaluation of indian classical raga using rapidminer. In *Soft Computing: Theories and Applications*, pages 97–106. Springer, 2018.

[8] Surendra Shetty and KK Achary. Raga mining of indian music by extracting arohana-avarohana pattern. *International Journal of Recent Trends in Engineering*, 1(1):362, 2009.

[9] Sathwik Tejaswi Madhusdhan and Girish Chowdhary. Tonic independent raag classification in indian classical music. 2018.

[10] Emiru Tsunoo, George Tzanetakis, Nobutaka Ono, and Shigeki Sagayama. Beyond timbral statistics: Improving music classification using percussive patterns and bass lines. *IEEE Transactions on Audio, Speech, and Language Processing*, 19(4):1003–1014, 2011.

[11] Mark Levy and Mark Sandler. Music information retrieval using social tags and audio. *IEEE Transactions on Multimedia*, 11(3):383–395, 2009.

[12] Yi Qin and Alexander Lerch. Tuning frequency dependency in music classification. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 401–405. IEEE, 2019.

[13] David Cope. Computer analysis of musical allusions. *Computer Music Journal*,

27(1):11–28, 2003.

[14] Jerome Barthélemy and Alain Bonardi. Figured bass and tonality recognition. In *ISMIR*. Citeseer, 2001.

[15] Roger B Dannenberg and Ning Hu. Pattern discovery techniques for music audio. *Journal of New Music Research*, 32(2):153–163, 2003.

[16] Tao Li and Mitsunori Ogihara. Detecting emotion in music. 2003.

[17] Matija Marolt. A mid-level melody-based representation for calculating audio similarity. In *ISMIR*, pages 280–285, 2006.

[18] Alexander Lerch. *An Introduction to Audio Content Analysis*. John Wiley & Sons, Inc., jul 2012.

[19] Michael Kassler. Toward musical information retrieval. *Perspectives of New Music*, pages 59–67, 1966.

[20] D. K. Hunter, M. H. M. Nizam, M. C. Chia, I. Andonovic, K. M. Guild, A. Tzanakaki, M. J. O'Mahony, L. D. Bainbridge, M. F. C. Stephens, R. V. Penty, and I. H. White. Waspnet: a wavelength switched packet network. *IEEE Communications Magazine*, 37(3):120–129, March 1999.

[21] Matthew J. Dovey. Overview of the OMRAS project: Online music retrieval and searching. *Journal of the American Society for Information Science and Technology*, 55(12):1100–1107, 2004.

[22] Andreas Kornstädt. The jring system for computer-assisted musicological analysis. In *ISMIR*, 2001.

[23] Franco Fabbri. A theory of musical genres: Two applications. In *Popular Music Perspectives: Papers from the First International Conference on Popular Music Research, Amsterdam*, pages 55–59, 1981.

[24] Perry R Cook. *Music, cognition, and computerized sound: An introduction to psychoacoustics*. The MIT Press, 1999.

[25] Peter Ahrendt. *Music genre classification systems*. PhD thesis, Ph. D. dissertation, Informatics and Mathematical Modelling, Technical , 2006.

[26] Laura Williams Macy. *Grove music online*. Macmillan Online, 2002.

[27] Yongwei Zhu and Mohan S Kankanhalli. Precise pitch profile feature extraction from musical audio for key detection. *IEEE Transactions on Multimedia*, 8(3):575–584, 2006.

[28] Yongwei Zhu, Mohan S Kankanhalli, and Sheng Gao. Music key detection for musical audio. In *11th International Multimedia Modelling Conference*, pages 30–37. IEEE, 2005.

[29] Juan Pablo Bello. Audio-based cover song retrieval using approximate chord sequences: Testing shifts, gaps, swaps and beats. In *ISMIR*, volume 7, pages 239–244.

Citeseer, 2007.

[30] Graham E Poliner, Daniel PW Ellis, Andreas F Ehmann, Emilia Gómez, Sebastian Streich, and Beesuan Ong. Melody transcription from music audio: Approaches and evaluation. *IEEE Transactions on Audio, Speech, and Language Processing*, 15(4):1247–1256, 2007.

[31] Daniel PW Ellis and Graham E Poliner. Identifying cover songs with chroma features and dynamic programming beat tracking. In *2007 IEEE International Conference on Acoustics, Speech and Signal Processing-ICASSP'07*, volume 4, pages IV–1429. IEEE, 2007.

[32] C. Lee, J. Shih, K. Yu, and H. Lin. Automatic music genre classification based on modulation spectral analysis of spectral and cepstral features. *IEEE Transactions on Multimedia*, 11(4):670–682, June 2009.

[33] David Moffat and Joshua D. Reiss. Perceptual evaluation of synthesized sound effects. *ACM Transactions on Applied Perception*, 15(2):1–19, apr 2018.

[34] H. Krim, P. Forster, and J.G. Proakis. Operator approach to performance analysis of root-MUSIC and root-min-norm. *IEEE Transactions on Signal Processing*, 40(7):1687–1696, jul 1992.

[35] Daniel PW Ellis. Classifying music audio with timbral and chroma features. 2007.

[36] G Emilia. *omez. Tonal Description of music audio signals.* PhD thesis, PhD thesis, Universitat Pompeu Fabra, 2006.(Cited on page 48).

[37] Shruti Sarika Chakraborty and Ranjan Parekh. Improved musical instrument classification using cepstral coefficients and neural networks. In *Methodologies and Application Issues of Contemporary Computing Framework*, pages 123–138. Springer, 2018.

[38] Alain Bonardi. Ir for contemporary music: What the musicologist needs. In *ISMIR*, 2000.

[39] Takuya Fujishima. Real-time chord recognition of musical sound: A system using common lisp music. *Proc. ICMC, Oct. 1999*, pages 464–467, 1999.

[40] Christine Senac, Thomas Pellegrini, Florian Mouret, and Julien Pinquier. Music feature maps with convolutional neural networks for music genre classification. In *Proceedings of the 15th International Workshop on Content-Based Multimedia Indexing*, page 19. ACM, 2017.

[41] Jean-Julien Aucouturier, Francois Pachet, et al. Music similarity measures: What's the use? In *ISMIR*, pages 13–17, 2002.

[42] H. Krim and M. Viberg. Two decades of array signal processing research: the parametric approach. *IEEE Signal Processing Magazine*, 13(4):67–94, jul 1996.

[43] David W. Shattuck and Richard M. Leahy. BrainSuite: An automated cortical

surface identification tool. In *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2000*, pages 50–61. Springer Berlin Heidelberg, 2000.

[44] S. Talwar, M. Viberg, and A. Paulraj. Blind estimation of multiple co-channel digital signals using an antenna array. *IEEE Signal Processing Letters*, 1(2):29–31, feb 1994.

[45] David Moffat and Joshua D Reiss. Perceptual evaluation of synthesized sound effects. *ACM Transactions on Applied Perception (TAP)*, 15(2):13, 2018.

[46] 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.

[47] N. Delprat, B. Escudie, P. Guillemain, R. Kronland-Martinet, P. Tchamitchian, and B. Torresani. Asymptotic wavelet and gabor analysis: extraction of instantaneous frequencies. *IEEE Transactions on Information Theory*, 38(2):644–664, mar 1992.

[48] John Saint Quattro, Alexander Varlamov, and Siu Ka Shan. Method and apparatus for making music selection based on acoustic features, August 2018. US Patent App. 15/593,309.

[49] Yu Shao and Christopher Bystroff. Predicting interresidue contacts using templates and pathways. *Proteins: Structure, Function, and Genetics*, 53(S6):497–502, 2003.

[50] G Komarasamy and Amitabh Wahi. An optimized k-means clustering technique using bat algorithm. *European Journal of Scientific Research*, 84(2):263–273, 2012.

[51] J.C. Mosher and R.M. Leahy. Recursive MUSIC: A framework for EEG and MEG source localization. *IEEE Transactions on Biomedical Engineering*, 45(11):1342–1354, 1998.

[52] RK Misra and SP Singh. Steady-state security analysis using artificial neural network. *Electric Power Components and Systems*, 32(11):1063–1081, 2004.

[53] Roger B. Dannenberg. Music representation issues, techniques, and systems. *Computer Music Journal*, 17(3):20, 1993.

[54] Joan Serra, Emilia Gómez, Perfecto Herrera, and Xavier Serra. Chroma binary similarity and local alignment applied to cover song identification. *IEEE Transactions on Audio, Speech, and Language Processing*, 16(6):1138–1151, 2008.

[55] Kayoko Yanagisawa, Ranniery Maia, and Yannis Stylianou. Multi-stream spectral representation for statistical parametric speech synthesis. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5160–5164. IEEE, 2016.

[56] Reena Rathee Jaglan, Rashid Mustafa, and Sunil Agrawal. Scalable and robust ann based cooperative spectrum sensing for cognitive radio networks. *Wireless Personal Communications*, 99(3):1141–1157, 2018.

[57] Keunwoo Choi, György Fazekas, Kyunghyun Cho, and Mark Sandler. The effects of noisy labels on deep convolutional neural networks for music tagging. *IEEE Transactions on Emerging Topics in Computational Intelligence*, 2(2):139–149, 2018.

[58] Aaron Van den Oord, Sander Dieleman, and Benjamin Schrauwen. Deep content-based music recommendation. In *Advances in neural information processing systems*, pages 2643–2651, 2013.

[59] Philippe Hamel and Douglas Eck. Learning features from music audio with deep belief networks. In *ISMIR*, volume 10, pages 339–344. Utrecht, The Netherlands, 2010.

[60] G. Rtsch, T. Onoda, and K.-R. Mller. Soft margins for adaboost. *Machine Learning*, 42(3):287–320, 2001.

[61] Michael I. Mandel, Graham E. Poliner, and Daniel P. W. Ellis. Support vector machine active learning for music retrieval. *Multimedia Systems*, 12(1):3–13, apr 2006.

[62] Timothy J Ross. *Fuzzy logic with engineering applications*. John Wiley & Sons, 2009.

[63] H. Hermansky and N. Morgan. RASTA processing of speech. *IEEE Transactions on Speech and Audio Processing*, 2(4):578–589, 1994.

[64] Changsheng Xu, Namunu Chinthaka Maddage, and Xi Shao. Automatic music classification and summarization. *IEEE transactions on speech and audio processing*, 13(3):441–450, 2005.

[65] Yuting Qi, Dehong Liu, David Dunson, and Lawrence Carin. Multi-task compressive sensing with dirichlet process priors. In *Proceedings of the 25th international conference on Machine learning - ICML '08*. ACM Press, 2008.

[66] Meinard Muller, Daniel PW Ellis, Anssi Klapuri, Ga"el Richard, and Shigeki Sagayama. Introduction to the special issue on music signal processing. *IEEE Journal of Selected Topics in Signal Processing*, 5(6):1085–1087, 2011.

[67] Lu Ren, David Dunson, Scott Lindroth, and Lawrence Carin. Dynamic nonparametric bayesian models for analysis of music. *Journal of the American Statistical Association*, 105(490):458–472, 2010.

[68] Tsukasa Endo, Shin-ichi Ito, Yasue Mitsukura, and Minoru Fukumi. The music analysis method based on melody analysis. In *2008 International Conference on Control, Automation and Systems*, pages 2559–2562. IEEE, 2008.

[69] Stefaan Lippens, Jean-Pierre Martens, and Tom De Mulder. A comparison of human and automatic musical genre classification. In *2004 IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 4, pages iv–iv. IEEE, 2004.

[70] Changsheng Xu, N.C. Maddage, and Xi Shao. Automatic music classification and summarization. *IEEE Transactions on Speech and Audio Processing*, 13(3):441–450, may 2005.

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

[72] Jing Lu, Wanggen Wan, Xiaoqing Yu, and Changlian Li. Music style classification using support vector machine. In *IET International Communication Conference on Wireless Mobile & Computing (CCWMC 2009)*. IET, 2009.

[73] Simone Scardapane, Danilo Comminiello, Michele Scarpiniti, and Aurelio Uncini. Music classification using extreme learning machines. In *2013 8th International Symposium on Image and Signal Processing and Analysis (ISPA)*. IEEE, sep 2013.

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

[75] Cory McKay and Ichiro Fujinaga. jsymbolic: A feature extractor for midi files. In *ICMC*, 2006.

[76] P Kirthika and Rajan Chattamvelli. A review of raga based music classification and music information retrieval (mir). In *Engineering Education: Innovative Practices and Future Trends (AICERA), 2012 IEEE International Conference on*, pages 1–5. IEEE, 2012.

[77] Y.M.D. Chathuranga and K.L. Jayaratne. Automatic music genre classification of audio signals with machine learning approaches. *GSTF International Journal on Computing (JoC Vol.3 No.2)*, 3(2), jul 2013.

[78] K. Aryafar, S. Jafarpour, and A. Shokoufandeh. Automatic musical genre classification using sparsity-eager support vector machines. In *Proceedings of the 21st International Conference on Pattern Recognition (ICPR2012)*, pages 1526–1529, Nov 2012.

[79] Fang-Fei Kuo and Man-Kwan Shan. A personalized music filtering system based on melody style classification. In *2002 IEEE International Conference on Data Mining, 2002. Proceedings.* IEEE Comput. Soc, 2002.

[80] Chong kai Wang, Ren-Yuan Lyu, and Yuang-Chin Chiang. An automatic singing transcription system with multilingual singing lyric recognizer and robust melody tracker. In *INTERSPEECH*, 2003.

[81] Matti Ryynanen, Tuomas Virtanen, Jouni Paulus, and Anssi Klapuri. Accompaniment separation and karaoke application based on automatic melody transcription. In *2008 IEEE International Conference on Multimedia and Expo*. IEEE, jun 2008.

[82] E. Pollastri and G. Simoncelli. Classification of melodies by composer with hidden

markov models. In *Proceedings First International Conference on WEB Delivering of Music. WEDELMUSIC 2001*. IEEE Comput. Soc.

[83] James G. Neal. Annual review of information science and technology. *The Journal of Academic Librarianship*, 21(5):401–402, sep 1995.

[84] Franois Pachet. Improving timbre similarity : How high s the sky ? 2004.

[85] Joe Futrelle and J. Stephen Downie. Interdisciplinary communities and research issues in music information retrieval. In *ISMIR*, 2002.

[86] Tzanetakis George, Essl Georg, and Cook Perry. Automatic musical genre classification of audio signals. In *Proceedings of the 2nd International Symposium on Music Information Retrieval, Indiana*, 2001.

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

[88] Ruben Hillewaere, Bernard Manderick, and Darrell Conklin. String methods for folk tune genre classification. In *ISMIR*, volume 2012, page 13th, 2012.

[89] Ruben Hillewaere, Bernard Manderick, and Darrell Conklin. String quartet classification with monophonic models. In *ISMIR*, pages 537–542, 2010.

[90] Ning Hu, R.B. Dannenberg, and G. Tzanetakis. Polyphonic audio matching and alignment for music retrieval. In *2003 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (IEEE Cat. No.03TH8684)*. IEEE.

[91] Darrell Conklin. Multiple viewpoint systems for music classification. *Journal of New Music Research*, 42(1):19–26, 2013.

[92] Ferda Ofli, Engin Erzin, Yücel Yemez, and A Murat Tekalp. Learn2dance: Learning statistical music-to-dance mappings for choreography synthesis. *IEEE Transactions on Multimedia*, 14(3):747–759, 2012.

[93] Erdem Ünal, Bariş Bozkurt, and M Kemal Karaosmanoğlu. A hierarchical approach to makam classification of turkish makam music, using symbolic data. *Journal of New Music Research*, 43(1):132–146, 2014.

[94] Joseph Picone, Tom Staples, Kazuhiro Kondo, and Nozomi Arai. Kanji-to-hiragana conversion based on a length-constrained n-gram analysis. *IEEE transactions on speech and audio processing*, 7(6):685–696, 1999.

[95] Yulin Ren and Dehua Li. Fast and robust wrapper method for n-gram feature template induction in structured prediction. *IEEE Access*, 5:19897–19908, 2017.

[96] Chang-Hsing Lee, Jau-Ling Shih, Kun-Ming Yu, and Hwai-San Lin. Automatic music genre classification based on modulation spectral analysis of spectral and cepstral features. *IEEE Transactions on Multimedia*, 11(4):670–682, 2009.

[97] Preeti Rao, Joe Cheri Ross, Kaustuv Kanti Ganguli, Vedhas Pandit, Vignesh Ishwar, Ashwin Bellur, and Hema A Murthy. Classification of melodic motifs in raga

music with time-series matching. *Journal of New Music Research*, 43(1):115–131, 2014.

[98] Hahn-Ming Lee, Chih-Ming Chen, Jyh-Ming Chen, and Yu-Lu Jou. An efficient fuzzy classifier with feature selection based on fuzzy entropy. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 31(3):426–432, 2001.

[99] Vilém Novák, Irina Perfilieva, and Jiri Mockor. *Mathematical principles of fuzzy logic*, volume 517. Springer Science & Business Media, 2012.

[100] Pedro Lucas, Enrique Pel, et al. Human-machine musical composition in real-time based on emotions through a fuzzy logic approach. In *2015 Latin America Congress on Computational Intelligence (LA-CCI)*, pages 1–6. IEEE, 2015.

[101] M. Holschneider, R. Kronland-Martinet, J. Morlet, and Ph. Tchamitchian. A real-time algorithm for signal analysis with the help of the wavelet transform. In *Wavelets*, pages 286–297. Springer Berlin Heidelberg, 1989.

[102] Hisao Ishibuchi, Tomoharu Nakashima, and Tadahiko Murata. Performance evaluation of fuzzy classifier systems for multidimensional pattern classification problems. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 29(5):601–618, 1999.

[103] Anssi Klapuri. Pattern induction and matching in music signals. In *International Symposium on Computer Music Modeling and Retrieval*, pages 188–204. Springer, 2010.

[104] Weishu Liu and Huchang Liao. A bibliometric analysis of fuzzy decision research during 1970–2015. *International Journal of Fuzzy Systems*, 19(1):1–14, 2017.

[105] Emanuele Pollastri and Giuliano Simoncelli. Classification of melodies by composer with hidden markov models. In *Web Delivering of Music, 2001. Proceedings. First International Conference on*, pages 88–95. IEEE, 2001.

[106] Haotian Xu and Zhijian Ou. Scalable discovery of audio fingerprint motifs in broadcast streams with determinantal point process based motif clustering. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 24(5):978–989, 2016.

[107] Alex S Park and James R Glass. Unsupervised pattern discovery in speech. *IEEE Transactions on Audio, Speech, and Language Processing*, 16(1):186–197, 2008.

[108] Xi Ma, Dong Wang, and Javier Tejedor. Similar word model for unfrequent word enhancement in speech recognition. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 24(10):1819–1830, 2016.

[109] Abhinav Sethy, Panayiotis G Georgiou, Bhuvana Ramabhadran, and Shrikanth Narayanan. An iterative relative entropy minimization-based data selection approach for n-gram model adaptation. *IEEE Transactions on Audio, Speech, and Language Processing*, 17(1):13–23, 2009.

[110] Ricard Marxer and Hendrik Purwins. Unsupervised incremental online learning and prediction of musical audio signals. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 24(5):863–874, 2016.

[111] Teemu Hirsimaki, Janne Pylkkonen, and Mikko Kurimo. Importance of high-order n-gram models in morph-based speech recognition. *IEEE Transactions on Audio, Speech, and Language Processing*, 17(4):724–732, 2009.

[112] Frédéric Bimbot, Roberto Pieraccini, Esther Levin, and Bishnu Atal. Variable-length sequence modeling: multigrams. *IEEE Signal Processing Letters*, 2(6):111–113, 1995.

[113] Manhung Siu and Mari Ostendorf. Variable n-grams and extensions for conversational speech language modeling. *IEEE Transactions on Speech and Audio Processing*, 8(1):63–75, 2000.

[114] Rui Chen, Gergely Acs, and Claude Castelluccia. Differentially private sequential data publication via variable-length n-grams. In *Proceedings of the 2012 ACM conference on Computer and communications security*, pages 638–649. ACM, 2012.

[115] Mehryar Mohri, Fernando Pereira, and Michael Riley. Weighted finite-state transducers in speech recognition. *Computer Speech & Language*, 16(1):69–88, 2002.

[116] Brian Roark, Murat Saraclar, and Michael Collins. Discriminative n-gram language modeling. *Computer Speech & Language*, 21(2):373–392, 2007.

[117] Daniel Crow and Barbara Smith. *DB_Habits: Comparing minimal knowledge and knowledge-based approaches to pattern recognition in the domain of user-computer interactions*. Ellis Horwood, 1992.

[118] Emilios Cambouropoulos. Towards a general computational theory of musical structure. 1998.

[119] Jia-Lien Hsu, Arbee LP Chen, and C-C Liu. Efficient repeating pattern finding in music databases. In *Proceedings of the seventh international conference on Information and knowledge management*, pages 281–288. ACM, 1998.

[120] Pierre-Yves Rolland. Discovering patterns in musical sequences. *Journal of New Music Research*, 28(4):334–350, 1999.

[121] David Meredith. Cosiatec and siateccompress: Pattern discovery by geometric compression. In *International Society for Music Information Retrieval Conference*, 2013.

[122] David Meredith. Music analysis and point-set compression. *Journal of New Music Research*, 44(3):245–270, 2015.

[123] Tom Collins, Andreas Arzt, Sebastian Flossmann, and Gerhard Widmer. Siarct-cfp: Improving precision and the discovery of inexact musical patterns in point-set representations. In *ISMIR*, pages 549–554, 2013.

[124] Daniel Müllensiefen and Klaus Frieler. Optimizing measures of melodic similarity for the exploration of a large folk song database. In *ISMIR*. Citeseer, 2004.

[125] Salha M Alzahrani, Naomie Salim, and Ajith Abraham. Understanding plagiarism linguistic patterns, textual features, and detection methods. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 42(2):133–149, 2012.

[126] Stanley F Chen and Ronald Rosenfeld. A survey of smoothing techniques for me models. *IEEE transactions on Speech and Audio Processing*, 8(1):37–50, 2000.

[127] W. Pedrycz and M. Song. Analytic hierarchy process (AHP) in group decision making and its optimization with an allocation of information granularity. *IEEE Transactions on Fuzzy Systems*, 19(3):527–539, June 2011.

[128] Z. Xu and H. Liao. Intuitionistic fuzzy analytic hierarchy process. *IEEE Transactions on Fuzzy Systems*, 22(4):749–761, Aug 2014.

[129] Peijia Ren, Zeshui Xu, and Huchang Liao. Intuitionistic multiplicative analytic hierarchy process in group decision making. *Computers & Industrial Engineering*, 101:513–524, 2016.

[130] Shahzad Faizi, Tabasam Rashid, Wojciech Salha M Alzahraniabun, Sohail Zafar, and Jaroslaw Watrobski. Decision making with uncertainty using hesitant fuzzy sets. *International Journal of Fuzzy Systems*, 20(1):93–103, 2018.

[131] Chandanpreet Kaur and Ravi Kumar. Classification of melodic structures using fuzzified n-gram matching scores. In *Fuzzy Systems (FUZZ-IEEE), 2016 IEEE International Conference on*, pages 685–690. IEEE, 2016.

[132] F Morchen, Alfred Ultsch, Michael Thies, and Ingo Lohken. Modeling timbre distance with temporal statistics from polyphonic music. *IEEE Transactions on Audio, Speech, and Language Processing*, 14(1):81–90, 2006.

[133] James Bergstra, Norman Casagrande, Dumitru Erhan, Douglas Eck, and Balázs Kégl. Aggregate features and a da b oost for music classification. *Machine learning*, 65(2-3):473–484, 2006.

[134] Michael Mandel and Dan Ellis. Song-level features and svms for music classification. In *In Proceedings of the 6th International Conference on Music Information Retrieval, ISMIR*, volume 5, 2006.

[135] Jialie Shen, John Shepherd, Bin Cui, and Kian-Lee Tan. A novel framework for efficient automated singer identification in large music databases. *ACM Transactions on Information Systems (TOIS)*, 27(3):18, 2009.

[136] Lie Lu, Dan Liu, and Hong-Jiang Zhang. Automatic mood detection and tracking of music audio signals. *IEEE Transactions on audio, speech, and language processing*, 14(1):5–18, 2006.

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

[138] Eric Allamanche. Content-based identification of audio material using mpeg-7 low level description. 01 2001.

[139] E. Benetos, M. Kotti, and C. Kotropoulos. Musical instrument classification using non-negative matrix factorization algorithms and subset feature selection. In *2006 IEEE International Conference on Acoustics Speed and Signal Processing Proceedings*. IEEE.

[140] Heng-Tze Cheng, Yi-Hsuan Yang, Yu-Ching Lin, I-Bin Liao, and Homer H. Chen. Automatic chord recognition for music classification and retrieval. In *2008 IEEE International Conference on Multimedia and Expo*. IEEE, jun 2008.

[141] George Tzanetakis, Randy Jones, and Kirk McNally. Stereo panning features for classifying recording production style. pages 441–444, 01 2007.

[142] Akhilesh K. Sharma and Prakash Ramani. Rigorous data analysis and performance evaluation of indian classical raga using RapidMiner. In *Advances in Intelligent Systems and Computing*, pages 97–106. Springer Singapore, nov 2017.

[143] Yannis Panagakis and Constantine Kotropoulos. Elastic net subspace clustering applied to pop/rock music structure analysis. *Pattern Recognition Letters*, 38:46–53, 2014.

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

[145] Pedro P Cruz-Alcázar and Enrique Vidal-Ruiz. Learning regular grammars to model musical style: Comparing different coding schemes. In *International Colloquium on Grammatical Inference*, pages 211–222. Springer, 1998.

[146] Wei Chai and Barry Vercoe. Folk music classification using hidden markov models. In *Proceedings of international conference on artificial intelligence*, volume 6. sn, 2001.

[147] Alessandro Moschitti and Roberto Basili. Complex linguistic features for text classification: A comprehensive study. In *European Conference on Information Retrieval*, pages 181–196. Springer, 2004.

[148] Pedro P Cruz-Alcázar, Enrique Vidal-Ruiz, and Juan C Pérez-Cortés. Musical style identification using grammatical inference: The encoding problem. In *Iberoamerican Congress on Pattern Recognition*, pages 375–382. Springer, 2003.

[149] Jun Li, Karen L Butler-Purry, Carl L Benner, and BD Russell. Selecting a fuzzy aggregation operator for multicriteria fault location problem. In *Power Systems Conference and Exposition, 2004. IEEE PES*, pages 1476–1482. IEEE, 2004.

[150] Yannis Panagakis, Constantine Kotropoulos, and Gonzalo R Arce. Non-negative

multilinear principal component analysis of auditory temporal modulations for music genre classification. *IEEE Transactions on Audio, Speech, and Language Processing*, 18(3):576–588, 2010.

[151] Zhouyu Fu, Guojun Lu, Kai-Ming Ting, and Dengsheng Zhang. On feature combination for music classification. In *Joint IAPR International Workshops on Statistical Techniques in Pattern Recognition (SPR) and Structural and Syntactic Pattern Recognition (SSPR)*, pages 453–462. Springer, 2010.

[152] Gao Huang, Shiji Song, Jatinder ND Gupta, and Cheng Wu. Semi-supervised and unsupervised extreme learning machines. *IEEE transactions on cybernetics*, 44(12):2405–2417, 2014.

[153] Jia Min Karen Kua, Vidhyasaharan Sethu, Phu Le, and Eliathamby Ambikairajah. The unsw submission to interspeech 2014 compare cognitive load challenge. In *Fifteenth Annual Conference of the International Speech Communication Association*, 2014.

[154] Loris Nanni, Sheryl Brahnam, Stefano Ghidoni, and Alessandra Lumini. Toward a general-purpose heterogeneous ensemble for pattern classification. *Computational intelligence and neuroscience*, 2015:85, 2015.

[155] Yandre MG Costa, Luiz S Oliveira, and Carlos N Silla Jr. An evaluation of convolutional neural networks for music classification using spectrograms. *Applied soft computing*, 52:28–38, 2017.

[156] Mehrdad Farrokhmanesh and Ali Hamzeh. A novel method for malware detection using audio signal processing techniques. In *2016 Artificial Intelligence and Robotics (IRANOPEN)*, pages 85–91. IEEE, 2018.

[157] Chandanpreet Kaur and Ravi Kumar. Study and analysis of feature based automatic music genre classification using gaussian mixture model. In *Inventive Computing and Informatics (ICICI), International Conference on*, pages 465–468. IEEE, 2017.

[158] Sukhbir S Kapoor. *Guru Granth Sahib-An Advance Study Volume-I.* Hemkunt Press, 2005.

[159] Anunaad Music Academy. Anunaad Music Academy, http://www.anunaadacademy.com/, 2019.

[160] F. Gouyon, A. Klapuri, S. Dixon, M. Alonso, G. Tzanetakis, C. Uhle, and P. Cano. An experimental comparison of audio tempo induction algorithms. *IEEE Transactions on Audio, Speech and Language Processing*, 14(5):1832–1844, sep 2006.

[161] Cory McKay, John Ashley Burgoyne, Jason Hockman, Jordan BL Smith, Gabriel Vigliensoni, and Ichiro Fujinaga. Evaluating the genre classification performance of lyrical features relative to audio, symbolic and cultural features. In *ISMIR*, pages

213–218, 2010.

[162] Cory McKay, Daniel McEnnis, and Ichiro Fujinaga. A large publicly accassible prototype audio database for music research. In *ISMIR*, pages 160–163, 2006.

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

[164] Kyaw Kyaw Htike. Forests of unstable hierarchical clusters for pattern classification. *Soft Computing*, 22(5):1711–1718, 2018.

[165] William Bruce Frakes and Ricardo Baeza-Yates. *Information retrieval: Data structures & algorithms*, volume 331. prentice Hall Englewood Cliffs, New Jersey, 1992.

[166] Ravi Sankar Sangam and Hari Om. An equi-biased k-prototypes algorithm for clustering mixed-type data. *Sādhanā*, 43(3):37, 2018.

[167] Michael L Seltzer, Bhiksha Raj, and Richard M Stern. A bayesian classifier for spectrographic mask estimation for missing feature speech recognition. *Speech Communication*, 43(4):379–393, 2004.

[168] Lawrence R Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989.

[169] Pallavi Purohit and Ritesh Joshi. A new efficient approach towards k-means clustering algorithm. *International Journal of computer applications*, 65(11), 2013.

[170] Alan Jose, S Ravi, and M Sambath. Brain tumor segmentation using k-means clustering and fuzzy c-means algorithms and its area calculation. *International Journal of Innovative Research in Computer and Communication Engineering*, 2(3):3496–3501, 2014.

[171] Madhu Yedla, Srinivasa Rao Pathakota, and TM Srinivasa. Enhancing k-means clustering algorithm with improved initial center. *International Journal of computer science and information technologies*, 1(2):121–125, 2010.

[172] V ROYNA DAISY and S NIRMALA. Stability-integrated fuzzy c means segmentation for spatial incorporated automation of number of clusters. *Sādhanā*, 43(3):40, 2018.

[173] Akhilesh K Sharma, Kamaljit I Lakhtaria, Avinash Panwar, and Santosh Vishwakarma. An analytical approach based on self organized maps (som) in indian classical music raga clustering. In *Contemporary Computing (IC3), 2014 Seventh International Conference on*, pages 449–453. IEEE, 2014.

[174] Mark Levy and Mark Sandler. Structural segmentation of musical audio by constrained clustering. *IEEE transactions on audio, speech, and language processing*, 16(2):318–326, 2008.

[175] Jonathan Foote. Automatic audio segmentation using a measure of audio novelty.

In *Multimedia and Expo, 2000. ICME 2000. 2000 IEEE International Conference on*, volume 1, pages 452–455. IEEE, 2000.

[176] Namunu C Maddage, Changsheng Xu, Mohan S Kankanhalli, and Xi Shao. Content-based music structure analysis with applications to music semantics understanding. In *Proceedings of the 12th annual ACM international conference on Multimedia*, pages 112–119. ACM, 2004.

[177] Dietmar Kunz. An orientation-selective orthogonal lapped transform. *IEEE Transactions on Image Processing*, 17(8):1313–1322, 2008.

[178] Vijaya V Chamundeeswari, Dharmendra Singh, and Kuldip Singh. An analysis of texture measures in pca-based unsupervised classification of sar images. *IEEE Geoscience and Remote Sensing Letters*, 6(2):214–218, 2009.

[179] Gilbert Strang. Linear algebra and its applications brooks. *Cole Thomson Learning Inc*, 1988.

[180] Jianqin Zhou and Ping Chen. Generalized discrete cosine transform. In *Circuits, Communications and Systems, 2009. PACCS'09. Pacific-Asia Conference on*, pages 449–452. IEEE, 2009.

[181] Emery Schubert, Sergio Canazza, Giovanni De Poli, and Antonio Rodà. Algorithms can mimic human piano performance: The deep blues of music. *Journal of New Music Research*, 46(2):175–186, 2017.

[182] Sovan Lek, Marc Delacoste, Philippe Baran, Ioannis Dimopoulos, Jacques Lauga, and Stéphane Aulagnier. Application of neural networks to modelling nonlinear relationships in ecology. *Ecological modelling*, 90(1):39–52, 1996.

[183] Vladimir Naumovich Vapnik. *Statistical learning theory*, volume 1. Wiley New York, 1998.

[184] Terry Harris. Credit scoring using the clustered support vector machine. *Expert Systems with Applications*, 42(2):741–750, 2015.

[185] Nazri Mohd Nawi, Walid Hasen Atomi, and MZ Rehman. The effect of data pre-processing on optimized training of artificial neural networks. *Procedia Technology*, 11:32–39, 2013.

[186] Bruce Benward. *Music in Theory and Practice Volume 1.* McGraw-Hill Higher Education, 2014.

[187] R Sudha, A Kathirvel, and RM Devanai Sundaram. A system of tool for identifying ragas using midi. In *Computer and Electrical Engineering, 2009. ICCEE'09. Second International Conference on*, volume 2, pages 644–647. IEEE, 2009.

[188] David L Davies and Donald W Bouldin. A cluster separation measure. *IEEE transactions on pattern analysis and machine intelligence*, (2):224–227, 1979.

[189] Hannah Daniel and A. Revathi. Raga identification of carnatic music using iter-

ative clustering approach. In *2015 International Conference on Computing and Communications Technologies (ICCCT)*. IEEE, feb 2015.

[190] Simon S Haykin, Simon S Haykin, Simon S Haykin, and Simon S Haykin. *Neural networks and learning machines*, volume 3. Pearson Upper Saddle River, NJ, USA:, 2009.

[191] Shan Suthaharan. *Machine learning models and algorithms for big data classification*. Springer, 2016.

[192] Alexandros Iosifidis and Moncef Gabbouj. Multi-class support vector machine classifiers using intrinsic and penalty graphs. *Pattern Recognition*, 55:231–246, 2016.

[193] Antoine Bordes, Seyda Ertekin, Jason Weston, and Léon Bottou. Fast kernel classifiers with online and active learning. *Journal of Machine Learning Research*, 6(Sep):1579–1619, 2005.

[194] EA Zanaty. Support vector machines (svms) versus multilayer perception (mlp) in data classification. *Egyptian Informatics Journal*, 13(3):177–183, 2012.

[195] Christoph Bergmeir and José M Benítez. On the use of cross-validation for time series predictor evaluation. *Information Sciences*, 191:192–213, 2012.

[196] Klaus Frieler. Generalized n-gram measures for melodic similarity. In *Data Science and Classification*, pages 289–298. Springer, 2006.

[197] Lloyd A Smith, Rodger J McNab, and Ian H Witten. Sequence-based melodic comparison: A dynamic programming approach. *Computing in musicology: a directory of research*, (11):101–118, 1998.

[198] Trevor Martin, Yun Shen, and Ben Azvine. Incremental evolution of fuzzy grammar fragments to enhance instance matching and text mining. *IEEE Transactions on Fuzzy Systems*, 16(6):1425–1438, 2008.

[199] Yi-Chan Wu and Homer H Chen. Generation of affective accompaniment in accordance with emotion flow. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 24(12):2277–2287, 2016.

[200] Yogesh Prabhakar Pingle and A. Bhagwat. Music therapy and data mining using indian ragas as a supplementary medicine. In *2015 2nd International Conference on Computing for Sustainable Global Development (INDIACom)*, pages 347–350, March 2015.

[201] Alan R Thrasher. *Qupai in Chinese Music: Melodic Models in Form and Practice*. Routledge, 2016.

[202] David Meredith, Kjell Lemström, and Geraint A Wiggins. Algorithms for discovering repeated patterns in multidimensional representations of polyphonic music. *Journal of New Music Research*, 31(4):321–345, 2002.

[203] Kjell Lemstrom. String matching techniques for music retrieval. 2002.

[204] David Meredith, Kjell Lemström, and Geraint A Wiggins. Algorithms for discovering repeated patterns in multidimensional representations of polyphonic music. *Journal of New Music Research*, 31(4):321–345, 2002.

[205] Corentin Louboutin and David Meredith. Using general-purpose compression algorithms for music analysis. *Journal of New Music Research*, 45(1):1–16, 2016.

[206] Gabriel D Cantareira, Luis Gustavo Nonato, and Fernando V Paulovich. Moshviz: A detail+ overview approach to visualize music elements. *IEEE Transactions on Multimedia*, 2016.

[207] Vladimir I Levenshtein. Binary codes capable of correcting deletions, insertions, and reversals. In *Soviet physics doklady*, volume 10, pages 707–710, 1966.

[208] Silviu Cucerzan and Eric Brill. Spelling correction as an iterative process that exploits the collective knowledge of web users. In *EMNLP*, volume 4, pages 293–300, 2004.

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

[210] Maarten Grachten, Josep-Lluıs Arcos, and Ramon López De Mántaras. Melodic similarity: Looking for a good abstraction level. *representations*, 2:7, 2004.

[211] Maarten Grachten, Josep Lluıs Arcos, and Ramon López De Mántaras. Melody retrieval using the implication/realization model. *MIREX http://www. music-ir. org/evaluation/mirex-results/article/s/similarity/grachten. pdf*, 2005.

[212] Mohamed M Mostafa and Nedret Billor. Recognition of western style musical genres using machine learning techniques. *Expert Systems with Applications*, 36(8):11378–11389, 2009.

[213] Mingchun Liu, Chunru Wan, and Lipo Wang. Content-based audio classification and retrieval using a fuzzy logic system: towards multimedia search engines. *Soft Computing*, 6(5):357–364, 2002.

[214] Joe Futrelle and J Stephen Downie. Interdisciplinary research issues in music information retrieval: Ismir 2000–2002. *Journal of New Music Research*, 32(2):121–131, 2003.

[215] Abdelwahab Hamam and Nicolas D Georganas. A comparison of mamdani and sugeno fuzzy inference systems for evaluating the quality of experience of hapto-audio-visual applications. In *Haptic Audio visual Environments and Games, 2008. HAVE 2008. IEEE International Workshop on*, pages 87–92. IEEE, 2008.

[216] Vaibhav Arora and Ravi Kumar. Probability distribution estimation of music signals in time and frequency domains. In *2014 19th International Conference on Digital Signal Processing*, pages 409–414. IEEE, 2014.

[217] Yusuf Yaslan and Zehra Cataltepe. Audio music genre classification using differ-

ent classifiers and feature selection methods. In *18th International Conference on Pattern Recognition (ICPR'06)*, volume 2, pages 573–576. IEEE, 2006.

[218] Indra Mohan Chakravarti, Radha Govira Laha, and Jogabrata Roy. Handbook of methods of applied statistics. *Wiley Series in Probability and Mathematical Statistics (USA) eng*, 1967.

[219] Stephan R Sain. The nature of statistical learning theory, 1996.

[220] Giovanni Fasano and Alberto Franceschini. A multidimensional version of the kolmogorov–smirnov test. *Monthly Notices of the Royal Astronomical Society*, 225(1):155–170, 1987.

[221] Frank J Massey Jr. The kolmogorov-smirnov test for goodness of fit. *Journal of the American statistical Association*, 46(253):68–78, 1951.

[222] Michael M Goodwin and Jean Laroche. A dynamic programming approach to audio segmentation and speech/music discrimination. In *2004 IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 4, pages iv–iv. IEEE, 2004.

[223] Janet Marques and Pedro J Moreno. A study of musical instrument classification using gaussian mixture models and support vector machines. *Cambridge Research Laboratory Technical Report Series CRL*, 4:143, 1999.

[224] Fang Zhou, Q Claire, and Ross D King. Predicting the geographical origin of music. In *2014 IEEE International Conference on Data Mining*, pages 1115–1120. IEEE, 2014.

[225] Michael Glodek, Martin Schels, and Friedhelm Schwenker. Ensemble gaussian mixture models for probability density estimation. *Computational Statistics*, 28(1):127–138, 2013.

[226] Jesper Højvang Jensen, Mads Græsbøll Christensen, and Søren Holdt Jensen. A tempo-insensitive representation of rhythmic patterns. In *2009 17th European Signal Processing Conference*, pages 1509–1512. IEEE, 2009.

[227] Andre Holzapfel and Yannis Stylianou. A scale transform based method for rhythmic similarity of music. In *2009 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 317–320. IEEE, 2009.

[228] Ugo Marchand and Geoffroy Peeters. The modulation scale spectrum and its application to rhythm-content analysis. 2014.

[229] Cory McKay and Ichiro Fujinaga. Improving automatic music classification performance by extracting features from different types of data. In *Multimedia Information Retrieval*, pages 257–266, 2010.