# 1. INTRODUCTION

## 1.1 Background

Distinguishing between music genres is a trivial task for human beings. A few seconds of music usually suffice to allow us to do a rough classification, such as identifying a song as rock or classical music [1]. The nebulous definitions and overlapping boundaries of genres makes reliable and consistent genre classification a difficult task for humans and computer alike. Traditional rules-based classification systems are severely limited by these factors as well as by the dynamic nature of genres [2]. The techniques used in this system are presented as alternative methods that can help to overcome these limitations.

The need for an effective automatic means of classifying music is becoming increasingly pressing as the number of recordings available continues to increase at a rapid rate. It is estimated that 2000 Compact Disks (CDs) a month are released for a wide distribution in Western countries alone. Software capable of performing automatic classifications would be particularly useful to the administrators of the rapidly growing networked music archives, as their success is heavily linked to the ease with which users can search for types of music on their sites. These sites currently rely on manual genre classifications, a methodology that is slow and inconsistent. Hence, we propose to build a system that is automatic, reliable, fast and consistent.

## 1.2 Objectives

Our primary objective is to develop a system that implements the automatic feature extraction and learning / pattern classification techniques that have the important benefit of being adaptable to a variety of other content-based (i.e. relating directly to and only to the music itself) musical analysis and classification tasks.

Our objectives can be pointed as:

* To develop a system that implements machine learning algorithms for fast and consistent classifications.
* To implement the principles and techniques of digital signal processing.
* To develop a system that can upgrade the current applications which feature the music genre classification and its implementation.
* To contribute the creation of more appropriate and specific music data warehouse.

## 1.3 Scope of the work

Music classiﬁcation is an interesting problem with many applications, from Drinkify (a program that generates cocktails to match the music) to Pandora to dynamically generating images that complements the music.

The automatic classiﬁcation of audio data according to music genres will aid the creation of music databases. It will also allow users to generate personal playlists on the ﬂy, where the user speciﬁes a general description such as 80s Synth-Pop, and the software does the actual ﬁle selection [1]. Furthermore, the features developed for automatic music genre recognition might also be useful in related ﬁelds such as similarity-based searching [2].

# 2. LITERATURE REVIEW

Music genre classification is not a new milestone in the era of technological development. Musical genre is used by retailers, libraries and people in general as a primary means of organizing music. Anyone who has attempted to search through the discount bins of a music store will have experienced the frustration of searching through music that is not sorted by genre. Listeners use genres to find music that they’re looking for or to get a rough idea of whether they’re likely to like a piece of music before hearing it. The music industry, in contrast, uses genre as a key way of defining and targeting different markets. The importance of genre in the mind of listeners is exemplified by research indicating that the style in which a piece is performed can influence listeners’ liking for the piece of the music. The types of features developed for a classification system could be adapted for other types of analyses by musicologists and music theorists. Taken in conjunction with genre classification results, the features could also provide valuable insights into the particular attributes of different genres and what characteristics are important in different cases. Automatic feature extraction and learning / pattern classification techniques have the important benefit of being adaptable to a variety of other content-based (i.e. relating directly to and only to the music itself) musical analysis and classification tasks, such as similarity measurements in general or segmentation. Systems could be constructed that, to give just a few examples, compare or classify pieces based on compositional or performance style, group music based on geographical / cultural origin or historical period, search for unknown music that a user might like based on examples of what he or she is known to like already, sort music based on perception of mood, or classify music based on when a user might want to listen to it (e.g. while driving, while eating dinner, etc.). Music librarians and database administrators could use these systems to classify recordings along whatever lines they wished. Individual users could use such systems to sort their music collections automatically as they grow and automatically generate play lists with certain themes. It would also be possible for them to upload their own classification parameters to search on-line databases equipped with the same classification software.

In the initial phase of our research, we studied several relevant papers. According to one of the papers that we referred, we concluded that the system developed can be fully modular, so that the genre hierarchy can be easily expanded in width and depth. It is also be possible to use alternative feature extraction methods and classifiers through plug-ins [1].

It turns out that music genres are very subjective. Individual tastes, the artist's previous work, along with the song time period can drive genre classification. Also from another research paper we followed, the algorithms mentioned in the following portion would be helpful in developing system which can classify the music genre based upon timbre [6]. Since genre is such a subjective classifier, additional research in unsupervised learning possibly could yield some very interesting results illustrating some kind of timbral grouping of music.

# 3. METHODOLOGY

Song genre tends to be quite subjective and genre tends to be applied in a general fashion to an artist as a whole. It would be more advantageous to group songs together that had the same timbre also known as the shape or color of sound [3]. The associated genre of music tends to be closely associated with timbre and is not necessarily an attribute of the artist. We try to address the problem of genre classification using only the audio content of a music file and techniques from machine learning.

We investigate various machine learning algorithms that include k-nearest neighbor (k-NN), k-means, multi-class Support Vector Machine (SVM) and neural networks to classify the music genres. One of the proficient algorithms includes Mel Frequency Cepstral Coefficients (MFCCs) for the application of machine learning algorithms as one of the characteristics feature.

## 3.1 Feature Vectors and Learning Algorithms

* MFCCs

MFCCs are a short-time spectral decomposition of an audio signal that conveys the general frequency characteristics important to human hearing. MFCCs are commonly used in the field of speech recognition. Recent research [4] shows that MFCCs are capable of capturing useful sound characteristics of music files as well. Our premise is that MFCC's contain enough information about the timbre of a song to perform genre classification. To compute these features, a sound file is subdivided into small frames of about 20 ms each and then MFCCs are computed for each of these frames. Since the MFCCs are computed over short intervals of a song, they do not carry much information about the temporal attributes of a song, such as rhythm or tempo.

The general process for converting a waveform to its MFCCs is described in Logan [2] and roughly explained by the following pseudo code:

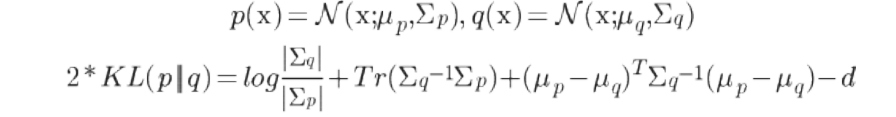
* Take the Fourier Transform of a frame of the waveform.
* Smooth the frequencies and map the spectrum obtained above onto the mel scale.
* Take the logs of the powers at each of the mel frequencies.
* Take the discrete cosine transform of the list of mel log powers, as if it were a signal.
* The MFCCs are the amplitudes of the resulting spectrum.

We compute N MFCC coefficients for all short duration frames of a wav file and store them in a F x N size matrix, where F is the number of frames.

### Song Level Modeling

Once the MFCC feature vectors for each frame of a song have been computed, an N-dimensional Gaussian is fit to this data (where N = 13)[5]. Hence all MFCC vectors of each song are modeled as coming from a multivariate Gaussian distribution. Using maximum likelihood estimation, the optimal Gaussian is calculated by simply taking the mean and covariance of the F x N MFCC matrix. Thus, ultimately each song is reduced to a 1 x N mean vector and an N x N covariance matrix. This greatly reduces the size of training data.

### Distance Function for Songs

Since each song is modeled as Gaussian probability distribution, it is natural to compute distance between songs using Kullback-Liebler divergence [5]. KL divergence is defined for any two probability distributions. In the case where both distributions are Gaussian of dimension d , it can be computed using the following closed form equation:

Where, p(x) and q(x) are two multivariate Gaussian distributions.

### Supervised Learning: k-NN

It is a non- parametric method used for classification and regression. In both cases, the input consists of the k closest training examples in the feature space. The neighbor are taken from a set of objects for which the class or the object property value is known.

In the implementation of this algorithm, during the training, we basically compute the Gaussian model for each song, and store these for all training samples for each genre. To guess the genre of a new song, we compute its distance, as defined above, with all samples in the training set, pick the k closest neighbors and assign it the predominant genre found amongst its nearest neighbors.

### Multi-Class SVM

SVM classifiers provide a reliable and fast way to differentiate between data with only two classes. In order to generalize SVMs to data falling into multiple classes (i.e. genres) a directed acyclic graph (DAG) of two-class SVMs trained on each pair of classes in the data. We then evaluate a sequence of two class SVMs and use a process of elimination to determine the output of our multi-class classifier.

## 3.2 Classification techniques

There are three main classification paradigms that can be used to perform automated classification:

1. Expert Systems: Expert Systems use pre-defined rules to process features and arrive at classifications.
2. Supervised Learning: This paradigm attempts to formulate classification rules by using machine learning techniques to train on model examples. Previously unseen examples are classified into one of the model categories using the patterns learned during training.
3. Unsupervised Learning: Unsupervised Learning cluster the data based on similarities that the systems perceive themselves. No model categories are used.

Expert systems are a tempting choice because known rules and characteristics of genres can be implemented directly. Many types of Western folk music, a great deal of non-Western music and Western popular music do not, in general, have the body of analytical literature that would be necessary to build an expert system. Although there are broad rules and guidelines that can be informally expressed about particular genres, it would be very difficult to design an expert system that could process rules that are often ill-defined and inconsistent across genres. A further problem is that new genres are constantly appearing and existing ones often change. Keeping a rule based system up to date would be a very difficult task.

Although unsupervised learning avoids the problems related to defining a set genre hierarchy and the categories produced might well be more accurate than human genre categories in terms of objective similarity, a genre classification system that uses its own genre categories would be of limited utility to humans who want to use genres that are meaningful and familiar to them.

Supervised learning is the best option. Such systems form their own rules without needing to interact with humans, meaning that the lack of clear genre definitions is not a problem. These systems can also easily be retrained to reflect changes in the genres being classified. There are a number of particular pattern classification techniques that can be used, including neural networks and k-NN.

## 3.3 Feature Extraction

In order to train a computer classifier, it is first necessary to extract features from musical recordings that can be given to the classifier as percepts. Simply giving the recording directly to a classifier would create an excess of information that would make the classification very slow and, quite likely, impossible. Extracting features from recordings and providing these to classifiers reduces the amount of information that must be processed and emphasizes aspects of recordings that are salient to the process of category discrimination.

The development of a large set of features is necessary to perceive the differences between any individual arbitrary pair genres coming from the large superset of genres in general. Although it is not feasible from a classification standpoint to deploy all of these features during a single classification operation, the use of a hierarchical taxonomy makes it possible to perform multiple classifications on different sub-trees of the hierarchy, each using specialized features. In other words, one could first make a coarse classification with a certain set of features, and then use different set of features to make finer classification.

# 4. EXPECTED OUTCOME

The system contains a GUI that can perform two tasks. One, it can train our system after we provide music clip with its genre. When training is complete, it displays a completion message. Two, it can test unknown music and hence find the genre to which it belongs. Hence when we provide our input i.e. music sample it first tests the sample with all the trained data and finally finds the best match among the trained genres. As it finds the best match, the GUI highlights one of the listed genres which the system depicts its best match.

# 5. PROJECT PLAN

# REQUIREMENTS

1. Software requirements:

* Integrated Development Environment (IDE) for java.

1. Resource requirements:

* High bandwidth internet service.
* Reports of past research on machine learning.
* Research reports on digital signal processing.
* API for controlling input and output of digital audio files and their manipulation.

# BIBLIOGRAPHY

[1]. Music Genre Recognition by Karin Koshina 2002

[2]. Music Genre Classification by Michael Haggblade, Yang Hong and Kenny Kao

[3].Sparse Multiview Methods for Classification of Musical Genre from Magnetoencephalography Recordings by Tom Diethe, Gabi Teodoru, Nick Furl and John Shawe-Taylor

[4]. Issues in Automatic Musical Genre Classification by Cory McKay

[5]. Supervised Learning in Genre Classification by Mohit Rajani and Luke Ekkizogloy

[6]. Song-level features and support vector machines for music classification by Mandel and Ellis

Table of Contents

[1. INTRODUCTION 1](#_Toc377466632)

[1.1 Background 1](#_Toc377466633)

[1.2 Objectives 1](#_Toc377466634)

[1.3 Scope of the work 2](#_Toc377466635)

[2. LITERATURE REVIEW 2](#_Toc377466636)

[3. METHODOLOGY 4](#_Toc377466637)

[3.1 Feature Vectors and Learning Algorithms 4](#_Toc377466638)

[ Song Level Modeling 5](#_Toc377466639)

[ Distance Function for Songs 5](#_Toc377466640)

[ Supervised Learning: k-NN 5](#_Toc377466641)

[ Multi-Class SVM 6](#_Toc377466642)

[3.2 Classification techniques 6](#_Toc377466643)

[3.3 Feature Extraction 7](#_Toc377466644)

[4. EXPECTED OUTCOME 8](#_Toc377466645)

[5. PROJECT PLAN 8](#_Toc377466646)

[6. REQUIREMENTS 9](#_Toc377466647)

[7. BIBLIOGRAPHY 9](#_Toc377466648)