Rock or not? This sure does.
[Category] Audio & Music
CS 229 Project Report

Anand Venkatesan*(anand95)*, Arjun Parthipan*(arjun777)*, Lakshmi Manoharan*(mlakshmi)*

1 Introduction

Music Genre Classification continues to be an interesting topic for study, given the ever-changing understanding of music genres and the extensive feature set we could build our machine learning models from. Successful implementation of genre classification can lead to more personalized music recommendations and well-defined music generation systems. As genres are a human abstraction, training machines to classify music based on genres is a non-trivial task.

For instance, we could argue that a genre is characterized by its rhythmic structure and instrumentation; but, there exist alternate arguments. Presently, there are more than 50 popular genres in music. Our objective is to identify an audio clip as belonging to a particular genre and providing song recommendations (from our data set) based on the identified genre.

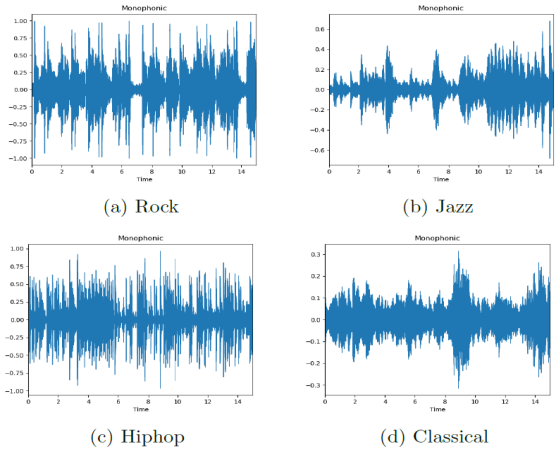
This report discusses (1) the literature pertaining to audio analysis and audio classification techniques (2) the data pre-processing method used to convert the audio files to matrix form for computation and analysis (3) the feature sets explored in this project (4) KL divergence as a distance metric to evaluate the similarity index (5) implementation of K-nearest neighbours, K-means clustering, Principal Component Analysis, multi-class poly-kernel Support Vector Machine and Deep Neural Network algorithms for genre classification (6) performance evaluation using confusion matrix, mean accuracy, recall, standard deviation, mutual information and random index score (7) inferences and conclusions (8) future work.

2 Literature Review

From our literature study, we gather that the classification can be done by representing the audio file in multiple ways. Previous studies show that this can be done using spectral and time domain analysis (waveform-based[1]) or Mel Frequency Cepstral Coefficients[2] (MFCC) to represent the audio data for further analysis. An interesting note in this regard is that some authors have attempted to translate the music genre classification problem into a text-classification problem based on lyrical features[3].

3 Data Preprocessing

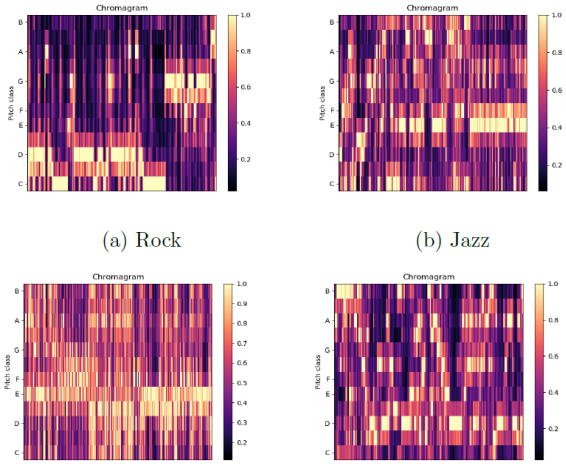
This project utilizes the GTZAN collection from MARSYAS [4], which consists of 100 tracks each for 10 genres, summing up to 1000 tracks in total. All the tracks are 22050Hz mono 16-bit audio files of 30s duration, which are represented in .au format. This project attempts to classify songs into four genres viz. Rock, Jazz, Hiphop, Classical.



(a) Rock (b) Jazz

(c) Hiphop (d) Classical

Figure 1: Amplitude Envelop Waveform



(a) Rock (b) Jazz

(c) Hiphop (d) Classical

Figure 2: Chromagram of the four genres

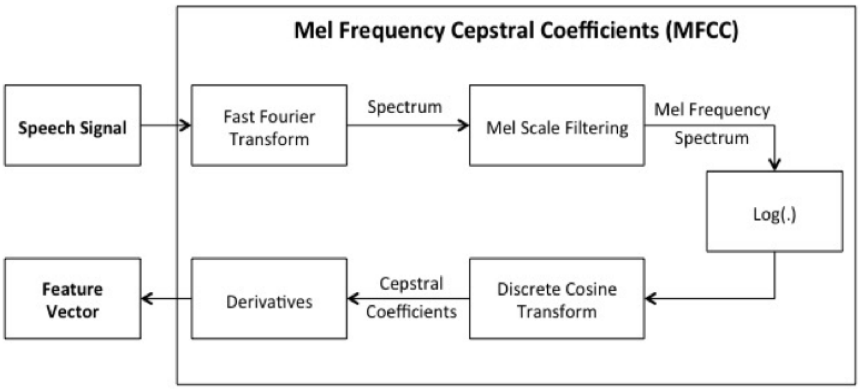


Figure 3: MFCC flowchart

Timbre is a perceived sound quality of a musical note that distinctly characterizes it [5]. We identified Mel Frequency Cepstral Coefficients (MFCC), traditionally used for speech recognition, to be representative of timbral features in music.

MFCCs are derived from a cepstral representation of the audio clip. Initially, we obtain the most significant portion of the song by clipping the middle portion (5-20 secs) of the track. The resulting 15s audio file is then split into 20ms frames, so that they do not carry information about temporal features such as rhythm. We then smoothen the edges by multiplying with a hamming window and apply Fast Fourier Transform to obtain the frequency components **(Figure 3)**.

With the frequency warping in mel scale, the frequency bands are equally spaced which approximates the human auditory systems response more closely than the linearly spaced frequency bands in the normal cepstrum. This mapping is done by calculating the mel scale triangular window coefficients and multiplying them with the frequencies. Finally, we retrieve the decorrelated frequency components by applying logarithm and computing the Discrete Cosine Transform.

4 Feature Set

This project utilizes two feature sets with each song track represented as
(1) A mean vector and covariance matrix containing the Mel-Frequency Cepstral Coefficients (MFCC)
(2) A flattened feature vector containing the mean vector and upper triangular elements of the covariance matrix.

On using n Mel-Frequency Ceptral Coefficients, we would be able to construct a n-element mean vector and a nxn covariance matrix. We conducted all of our experiments using 15 MFCCs, and repeated the same with 20 MFCCs. The flattened feature vector formed using n MFCCs would then contain (n(n+1)/2 + n)-elements.

5 Distance Metric

Given the non-scalar nature of attributes obtained from the MFCC processing, we could not use the conventional Euclidean distance metric to measure similarity between song tracks. Recent research identifies Kullback-Leibler divergence *(also called the relative entropy)* to be analogous to a distance metric for this purpose [6]. As KL Divergence statistically measures how one distribution diverges from the second, lower the value of the KL Divergence, closer are the two distributions and vice versa. If p(x) and q(x) represent two Multivariate Gaussian Distributions, with mean vectors and covariance matrices computed from the MFCC matrix of each song and d is the number of cepstral coefficients considered, the KL divergence is given by:

$$\begin{aligned} 2KL(p||q) = log\frac{|\Sigma_q|}{|\Sigma_p|} + tr(\Sigma_q^{-1}\Sigma_p) \\ +(\mu_p - \mu_q)^T\Sigma_q^{-1}(\mu_p - \mu_q) - d \end{aligned} \tag{1}$$

We know that the distance from p(x) to q(x) is the same as the distance from q(x) to p(x). Since the distance D is symmetric, we have

$$\begin{aligned} D_{KL}(p,q) &= KL(p||q) + KL(q||p) \\ &= 2KL(p||q) \end{aligned} \tag{2}$$

6 Classification Methods

6.1 K-nearest neighbours

K-Nearest Neighbours is a non-parametric classifier that classifies a given test point based on the majority class of its k-nearest training points. We found this algorithm to be vulnerable to high dimensional inputs. On considering more number of MFCC features, we observed a decrease in effectiveness of kNN.

6.2 K-Means

K-means is an unsupervised learning algorithm that involves grouping data into 'cohesive' clusters. Initially, the centroids of the clusters are chosen to be random songs from the training set, represented by their mean vectors and covariance matrices. Each song is then assigned to the cluster with the nearest centroid as computed using KL Divergence. The mean vector and the covariance matrix of each cluster centroid are then updated to the mean of the mean vectors and covariance matrices of the songs in that cluster.

6.3 Support Vector Machine

Support Vector Machine (SVM) is a discriminative classifier which is defined by a separating hyperplane which categorizes new data based on the labeled training samples [7]. Here, we have used multi-class SVM to obtain the optimal class boundaries between different genres of music. A directed acyclic graph based two class SVM trained on each pair of genres is implemented and is shown in **Figure 4**. A process of elimination was used to determine the output of the poly-kernel SVM.

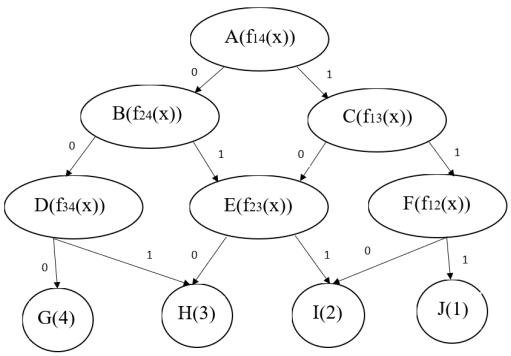


Figure 4: Two-class SVM

6.4 Deep Neural Network

Neural networks are built off the model of neurons present in the human brain. Neural networks are typically stacked layers of interconnected 'neurons' or nodes each having an activation function. The inputs are then processed through this system of weighted connections, allowing us to define non-linear relationships between the input and the dependent variable. We hypothesized that a non-linear combination of the MFCCs would be able to better model the features that characterize each genre. We hence implemented a 2-layer deep neural network with 135 and 124 neurons in the first and second hidden layers respectively.

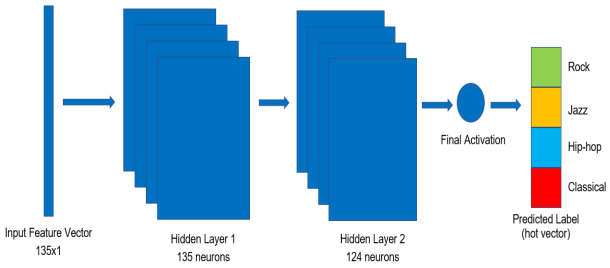
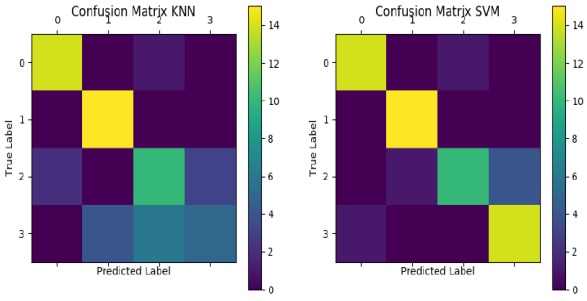


Figure 5: Neural Network

7 Results

We carried out two experiments considering (1) 20 MFCCs for each track (2) 15 MFCCs for each track (by discarding the coefficients corresponding to the higher frequencies). Additionally, we repeated the experiments using the flattened feature feature for 15, 20 MFCCs as described in **Section 4**. In general, we observed that the algorithms perform better for the smaller feature set. All results shown below correspond to the 15-MFCC feature set.



(a) K-NN (b) SVM

Figure 6: Confusion Matrix

The resulting models were evaluated both qualitatively and quantitatively: (1) qualitatively, by listening to the input audio clip and concluding

the relevant music genre (2) quantitatively, by using the following Figures of Merit to evaluate the model: Mean Accuracy (proportion of the number of correct trials to the total number of trials of the system), Recall (proportion of the number of correct trials of the system to the total number of a specific input label) accompanied by Standard Deviation and Confusions Matrix, Mutual Information Score and Random Index Score.

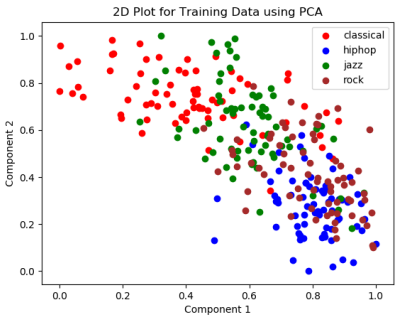


Figure 7: Two Dimensional Plot of Training Data

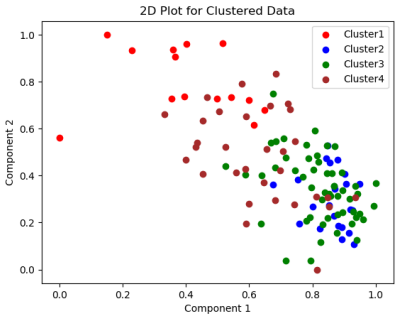


Figure 8: 2D Plot for Clustered data - 4 Genres

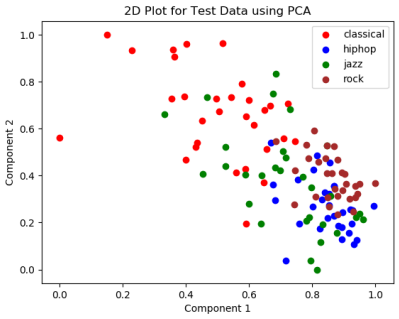
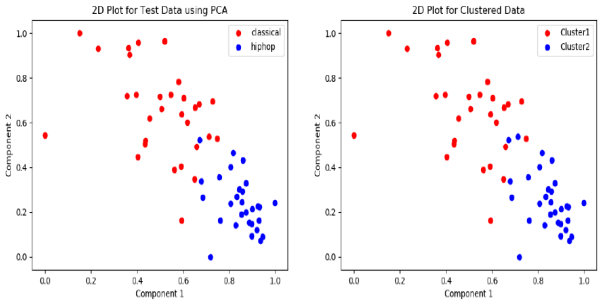


Figure 9: 2D Plot for Test Data - 4 Genres

We split the data set into training (70%), dev (15%) and test (15%) sets and obtained the optimal K value (K=8) by using the k-fold cross validation technique. On running the kNN algorithm using KL Divergence, we were able to obtain good results as tabulated in **Table 1**. The multi-class poly-kernel SVM algorithm using the KL Divergence distance metric yielded even better accuracy and recall as tabulated in **Table 2**. On using the flattened feature vector coupled with the Euclidean distance metric, there was a significant drop in performance for both the kNN and SVM models, indicating that the Euclidean distance does not accurately model the similarity between audio tracks.



(a) Test Data (b) Clustered Data

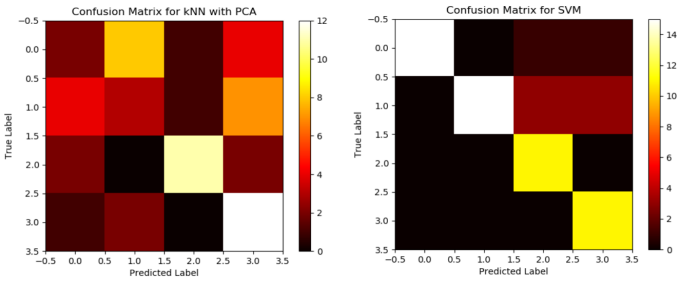
Figure 10: Two Dimensional Plot

TABLE 1: KNN Evaluation

Genres	Accuracy in %	Recall in %	Standard Deviation
Rock	78.3	33.3	1.30
Jazz	80	66.67	0.65
Hiphop	93.33	100	0
Classical	95	93.33	0.13

TABLE 2: SVM Evaluation

Genres	Accuracy in %	Recall in %	Standard Deviation
Rock	96.67	93.33	0.13
Jazz	98.33	100	0
Hiphop	90	66.67	0.65
Classical	91.67	93.33	0.13



(a) KNN (b) SVM

Figure 11: Confusion Matrix for flattened feature

We projected the training data onto a 2D space using Principal Component Analysis (PCA) to visualize clusters in the data set. As seen in **Figure 7**, we observed a significant overlap among the four genres. Consequently, K-means did not perform as well as kNN. We further attempted to run K-means for two-genre classification by considering all genre-pairs in the data set and were

able to achieve near-perfect classification. For the classical-hiphop genre pair, we obtained a Mutual Information Score of 0.89 and Random Index Score of 0.93, with well-defined clusters as captured in **Figure 10**. Given the overlap among the four genres chosen, SVM performed better than both kNN and K-means, as expected.

TABLE 3: NN Evaluation

Genres	Accuracy in %	Recall in %	Standard Deviation
Rock	73.33	87.5	0.51
Jazz	80	73.67	0.39
Hiphop	93.33	92.3	0.13
Classical	93.33	91.67	0.13

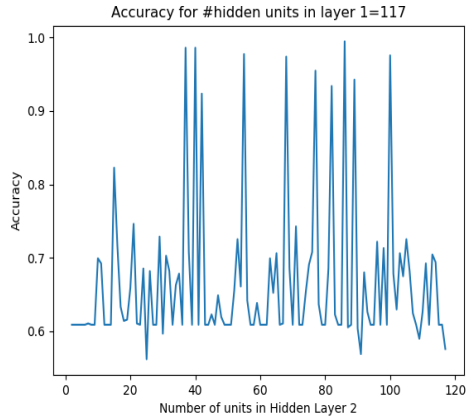


Figure 12: Accuracy vs #Hidden Units in Layer 2

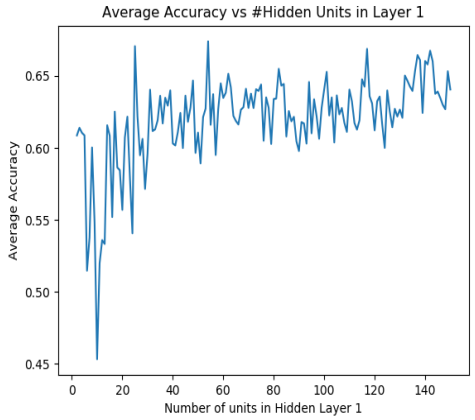


Figure 13: Avg Accuracy vs #Hidden Units in Layer 1

The 2-layer deep neural network was implemented using the flattened feature set consisting of 52 principal components. The optimal number of principal components was empirically determined. We also iterated for a different range of values to determine the optimal number of neurons in each hidden layer. Additionally, we experimented with the logistic and ReLu functions for activation of both the hidden layers, and found the ReLu activation to perform better than the logistic activation. As depicted in **Figure 12** and **Figure 13**, we found that the choice of $< 135, 124 >$ provided the best results. Despite the information loss incurred as a result of flattening and further dimensionality reduction using PCA, the implemented neural network model provided satisfactory results as tabulated in **Table 3**.

We built a recommendation engine, on top of the kNN classifier that uses the KL Divergence as a distance metric. The recommender system provides 10 track recommendations for a given audio track, based on genre similarity scores. On an average, we observed that it provides 8 out of 10 similar track recommendations.

8 Conclusion

We applied various supervised and unsupervised machine learning algorithms including kNN, K-means, Support Vector Machines and Deep Neural Networks to solve the music genre classification problem. We conducted all of our experiments using different feature sets and distance metrics including KL Divergence and Euclidean Distance coupled with Principal Component Analysis.

In addition, we built a recommender system that provides 10 track recommendations based on similarity scores obtained from the k-nearest neighbors algorithm. In conclusion, we observed the KL divergence distance metric to better capture the similarity between audio tracks. As a direct consequence, the kNN and SVM classifiers coupled with KL divergence outperform their counterparts and neural network models built using the Euclidean distance metric.

9 Next Steps

We plan to extend this work along the following lines in future:

- Explore neural network models that would enable incorporating the KL Divergence distance metric
- Extend the existing implementations using modified feature sets that include chroma frequencies, spectral centroids, spectral rolloffs and zero-crossing rates.

10 References

[1] Pontikakis, Charles Tripp Hochak Hung Manos. "Waveform-Based Musical Genre Classification."
[2] Fu, A., Lu, G., Ting, K.M., Zhang, D.. A Survey of Audio-Based Music Classification and Annotation IEEE Transactions on Multimedia. http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5664796&tag=1 [3] Bou-Rabee, Ahmed, Keegan Go, and Karanveer Mohan. "Classifying the Subjective: Determining Genre of Music From Lyrics." (2012).
[4] Marsyas. "Data Sets" http://marsysas.info /download/data _sets.
[5] De Poli and Prandoni, Sonological Models for Timbre Characterization, http://lcavwww.epfl.ch / prandoni/ documents/timbre2.pdf
[6] Mandel, M., Ellis, D.. Song-Level Features and SVMs for Music Classification http://www.ee.columbia.edu/dpwe/pubs/ismir05-svm.pdf.
[7] Chen, P., Liu, S.. An Improved DAG-SVM for Multi-class Classification http://ieeexplore.ieee.org/stamp/stamp.jsp? arnumber=0566976.

11 Contribution

We began with our literature review, with each of us reading several research papers pertaining to the topic and sharing our findings in Week 1. We identified GTZAN as the dataset for our project, and formulated a research plan. Each of us extensively read about one algorithm each, and practised a peer-teaching approach to share our progress and findings. This enabled us to collaboratively contribute to the codebase hosted on GitHub. Overall, we view this project as a product of collaborative efforts from all three of us.