MAT 167 Report: First Draft

JP Maestas

September 2025

Introduction

In an effort to dampen the use of illicit data scraping and API queries to train AI models, many organizations have removed these legal protocols altogether. One of those organizations is Spotify, which, as of recently, has disabled many features that allow users to obtain audio analysis from songs they query. This feature is the backbone for Spotify's recommendation algorithms, and its depreciation has prompted developers, including possibly myself, to obtain this information from illicit means.

Dataset

The data set which I chose to use for this project is courtesy of the Free Music Archive. I chose the small (7.2 Gb) dataset which consists of 8000 individual 30 second MP3 tracks. When we open one of these files, it is read within the time domain. That means that we perceive the signal's amplitude as its intensity (loudness) over time. However, if we use Fourier Transformation, we can decompose this signal into the frequency domain. This allows us to obtain core information about the pitch of the music. This is generally encoded into the chromatic scale. For the fourth octave, the one whose first note contains "Middle C", the mapping from notes to frequency range is roughly

C, C♯/D♭, D, D♯/E♭, E, F, F♯/G♭, G, G♯/A♭, A, A♯/B♭, B, C.

↓

261 277 293 311 329 349 369 392 415 440 466 493

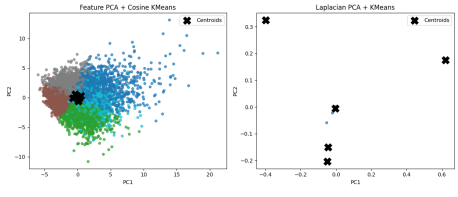

Figure 1:

Methods

When we listen to music, the frequency component of the sound waves encodes how we perceive the noise. These sound waves are oscillations of matter which we perceive over time. We can describe and transform these waves as an approximation of sinusoidal components to change the data from temporally encoded amplitude onto a basis of frequency.

Let ω denote angular frequency. This is the rate at which the wave rotates. Separately, t denotes time, and $i = \sqrt{-1}$. The significance of this term is that it allows us to encode a separate dimension (the imaginary axis) into our approximation. This is significant because sound waves are inherently three dimensional objects. So, we use the relation $e^{ix} = cos(x) + isin(x)$. With all of this information, we can finally decompose our signal into the frequency domain. This is given by

$$F(j\omega) = \int_{\infty}^{\infty} f(t)e^{-i\omega t}dt$$

However, although sound waves are continuous, we can only encode discrete data into a computer's memory. Therefore, we will approximate this transformation using Discrete fourier Transformation (DFT). That is,

$$F(j\omega) = \sum_{k=1-\frac{N}{2}}^{\frac{N}{2}} f(k)e^{\frac{-i\omega t}{N}}$$

Then, we use MFCC (Mel-Frequency Cepstral Coefficient)

$$Y_t[m] = \sum_{k=1}^{N} W_m[k]\,|X_t[k]|^2$$

where k : DFT bin number $(1, \ldots, N)$
m : mel-filter bank number $(1, \ldots, M)$

in order to create feature vectors P_v of each instance such that

$$P_v \in \mathbb{R}^{\omega t \times 1}$$

then, we will create an augmented matrix of each instance. We will call this matrix J such that

$$J = \begin{bmatrix} P_{v_1} & P_{v_2} & \dots P_{v_m} \end{bmatrix}$$

where

$$J \in \mathbb{R}^{\omega t \times m}$$

Since our data has very high dimensionality, we will use PCA and the k-means algorithm to find k clusters within our data that reduce their euclidean space to some arbitrary centroid. This will allow us to group features based on their components. We can create a mapping from features to these clusters in order to further use the algorithm as a reccomendation algorithm. Based on time constarints, however, I have not been able to map these yet. Separately, to compare to modern literature on clustering audio data, we will use Spectral Clustering. This process begins by computing a Similarity Matrix of our data. This is a matrix such that

$$JP = PB$$

or equivalently,

$$J = PBP^{-1}$$

We note that this is similar, but not equivalent to, to the eigendecomposition of J. From here, we can compute the non-normalized Laplacian matrix. This is

$$L = D - B$$

where

$$D = \begin{bmatrix} d_1 & 0 & \dots & 0 \\ 0 & d_2 & & \vdots \\ \vdots & & \ddots & 0 \\ 0 & \dots & 0 & d_k \end{bmatrix}$$

and

$$d_i = \sum_{i}^{k} B_{ij}$$

This matrix denotes the number of edges connected to each node in a graph. Thus, our matrix L contains condensed information of the signal instances into a matrix who we can further perform clustering on. For simplicty's sake, we will use the k-means algorithm.

Result

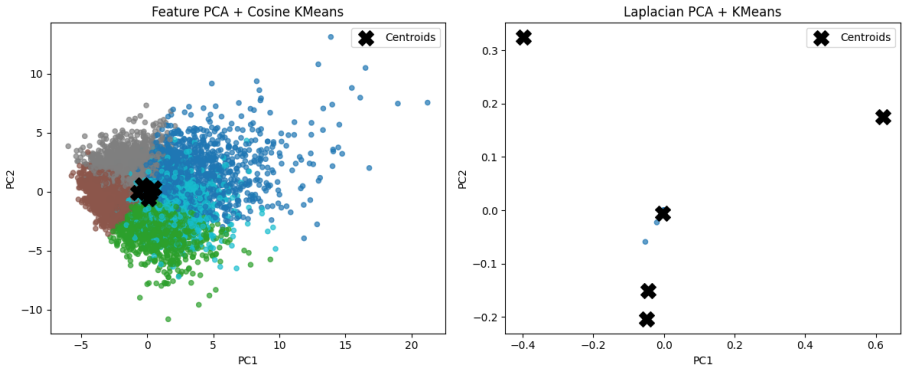

Figure 2: K Means C lustering and PCA on feature data

Once we have performed the k-means algorithm optimized by the elbow method, we categorized the songs into discrete subcategories. From this, we can create a set of graphs using the Networkx library to display which songs are connected by genre classification.