STA 142B Final Project: Methodology

Andrew Robertson

Truc Le

Aaron Mui

Siddharth Reddy Chandagari Venkata

Introduction

The task of implementing machine learning techniques to automate music genre classification based on audio contents has not been an easy endeavor[2]. Although Machine Learning has been successful in performing text classification and retrieving documents best fit to the user's specifications [1], there has been challenges to implementing Machine Learning techniques to perform music genre classification. This is in part due to the complexity of the audio signals [2]; where a song can be a mixed of multiple genre, thus, making it difficult to classify the song into a distinct genre. In addition, it is usually not possible to input the raw audio files into the classification model, therefore, the audio files are transformed into more compact forms containing a subset of features extracted from the original audio files.
The objective of this project is to use Machine Learning algorithms to cluster a set of unknown audio files and evaluate the performances of several audio clustering models to identify the best model. The audio data set contains 90 unlabeled mp3 files that are titled from 0 to 89, with each audio file being 30 seconds long. This project is outlined as follows: single dimension feature extraction followed by multi-dimensional feature extraction, dimensionality reduction using manifold learning, clustering the unlabeled mp3 files using different clustering approaches, and assessing the clustering model with the best cluster results.

Feature Extraction

For the purpose of feature extraction, we began our project by employing the following single-dimensional features: Spectral centroids, Spectral flatness, Spectral roll-off, Spectral bandwidth, Zero-crossing rate, and Root mean square. Furthermore, we employed the following multi-dimensional features: Constant-Q chromagram, MFCC, Tonnetz and Spectral contrast. Upon reducing the multi-dimensional features, we used summary statistics to extract the vital information from our feature space and represent the audio files. We chose the summary statistics: mean, variance, skewness, kurtosis, minimum and maximum. However, upon plotting our results, we found that the points along our dataset fell on a curve, indicating that we needed to engineer our feature space further.

For the next iteration of our model, we decided to standardise our data using the StandardScaler() function in the sci-kit learn package. Using a correlation matrix, we dropped the features with high correlation to reduce redundancy. Additionally, we streamlined our summary statistics process by only employing Mean, Standard Deviation and Maximum. Upon reviewing our results, we found that we obtained more discernible clusters using certain clustering techniques, but our data still appeared to lie along a curve.

Upon doing some feature exploration and research, we decided to drop a significant portion of our feature space and retain the following features: Spectral centroid, Spectral roll-off, Zero-crossing rate, MFCC, Constant-Q chromagram and Tonnetz. Additionally, using the hpss() function in the librosa package, we separated our audio files' harmonic and percussive features and utilised only the harmonic data for spectral analysis. Finally, upon refining our feature space, standardising our data and dropping features with high correlation, we obtained discernible clusters with a statistically significant increase in accuracy based on tests we derived for this project.

Dimension Reduction

When obtaining features using librosa's feature extraction module, rather than calculating the features across the entire song, it calculates them between frames. This is an issue because we want the feature to describe the entirety of the song. In order to work around this we decided to aggregate the across the frames using their mean, max, and standard deviation. There were two cases we needed to deal with: single-dimensional features and multi-dimensional features.

Single-Dimension Features. Depending on your parameters, each song would have around n frames, each frame having a distinct value for a certain feature. For one song, a feature like zero-crossing rate would have size $1 \times n$, a zero-crossing rate value for each frame. Since we used three single-dimensional features, our resulting matrix had size $n \times 90 \times 3$, with n frames, 90 songs, and 3 features. Taking the mean, max, and standard deviation across all frames of each song and feature led us to a $3 \times 90 \times 3$ matrix. Finally we

reshaped our matrix to two dimensions by taking each summary statistic and horizontally stacking them, giving us a final matrix size of 90×9.

Multi-Dimension Features. When extracting a multi-dimensional feature such as MFCC, it would have shape $13 \times n$, 13 MFCC values for each frame. The same can be said for other multi-dimensional features. Rather than using another dimensional reduction technique to reduce the 13 values into 1 or 2, we decided to keep every value for the sake of better clustering results. After extracting the three features, our initial matrix size was $n \times 90 \times 31$, where 31 is the sum of values returned per feature for a single frame – 13 for MFCC, 12 for constant-Q chromagram, and 6 for tonnetz. Again, using the three summary statistics, we aggregated over the frames which resulted in a $3 \times 90 \times 31$ matrix, which was reshaped into our final matrix with size 90×93.
After the two matrices were completed, we combined them resulting in a 90×112 matrix.

Clustering

The following two clustering methods were considered when building the clustering model, they are as follow: Agglomerative Hierarchical clustering and Mean-Shift clustering. We constructed the Agglomerative Hierarchical clustering model using the euclidean distance to compute the complete linkage between the observations, in addition, we gave the model the constraint of only finding 3 unique clusters within the inputted data set. For the Mean-Shift clustering model, we set the bandwidth size to be different depending on which method was used to perform dimension reduction on the inputted data. The different bandwidth sizes were 0.05, 0.026, or 25 depending on whether the data underwent dimension reduction using either the LLE, Spectral embedding, or tSNE methods respectively.

Model Assessment

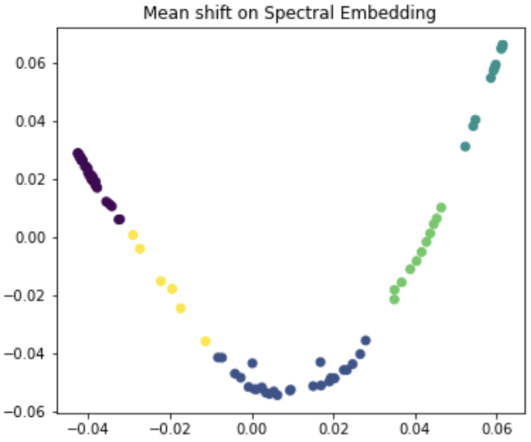
When determining the performance of our models, our choice of features was the most important consideration, followed by dimension reduction technique, then clustering.

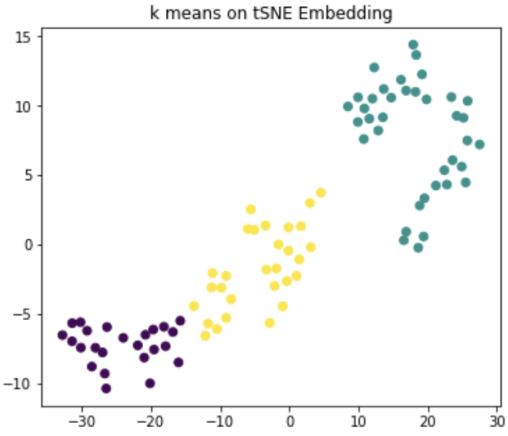
Features. Our feature space was the one piece of this project that changed the most as our team progressed. In some sense, the right feature space is a necessary condition for meaningful clustering. Though the project was unstructured, we considered the physical interpretation of our features first and foremost which led us to our final feature space as presented in our .ipynb. Our imperfect feature space, without a doubt was and still is the biggest bottleneck in our models' performance.

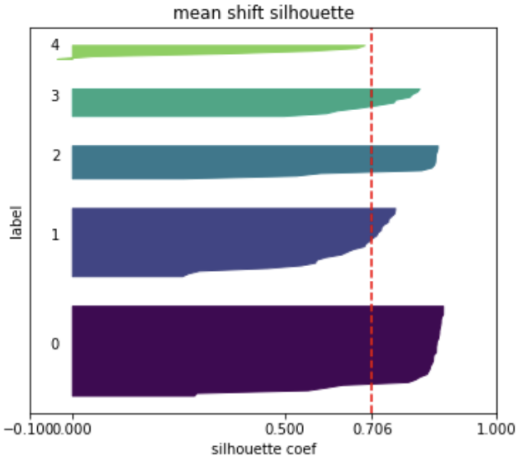
Dimension Reduction and Clustering. First, I'll note we fixed our embedding to be two dimensional for every method we tried. This choice gave us the ability to assess our embedding visually, but could be a problem if our data truly lie on a higher dimensional surface. Once we constructed embeddings, we would cluster with various methods and various hyperparameters, creating a feedback loop where we could tune the hyperparameters for both the clustering and dimension reduction to improve a few key metrics, namely, our silhouette plot, listening a few songs, and visually assessing the geometry of our embedding. These three metrics helped us close the loop with feature selection, allowing us to tweak our feature space then look at new embeddings/clusterings. I should also note that when we compared different dimension reduction and clustering techniques using the same feature space, we often found that if the feature space was held constant, moth methods would converge to one another. However, we found that when the feature space changed, this would create wildly different results. To choose among these different feature spaces, we leaned on listening to songs as the best metric to compare them. Thus, that is why we chose Local Linear Embedding and hierarchical clustering as our final labels. In the figures below, one can see what we considered our best clustering for each iteration of our feature space.

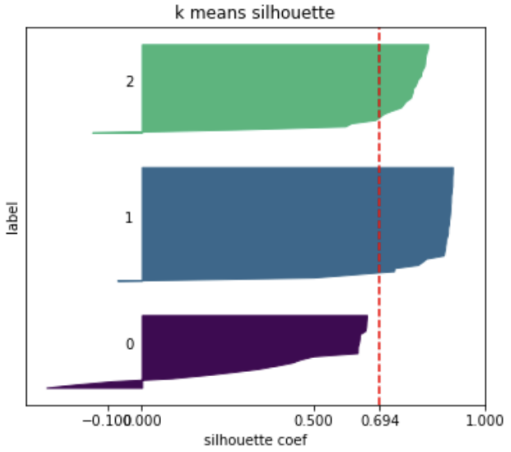
Figures

Feature Space 1 Clustering

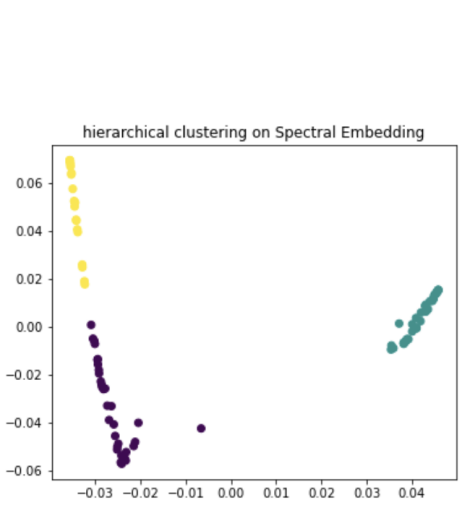


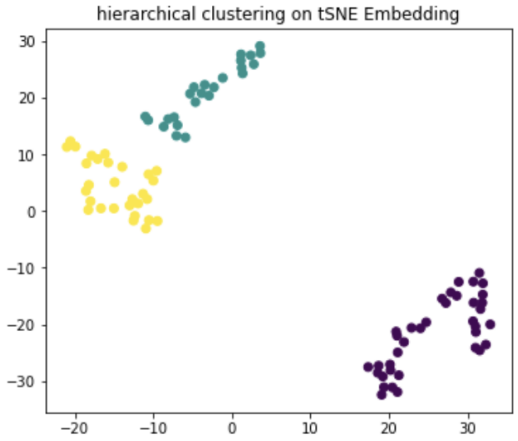


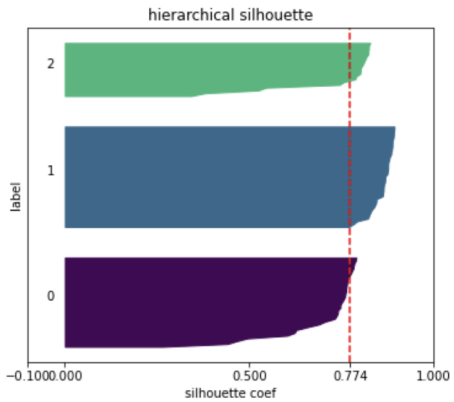


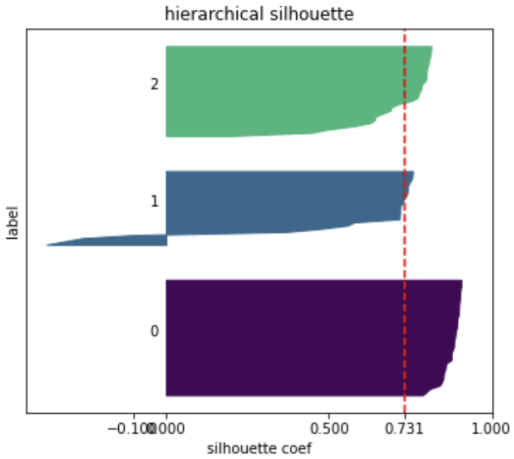


Feature Space 2 Clustering

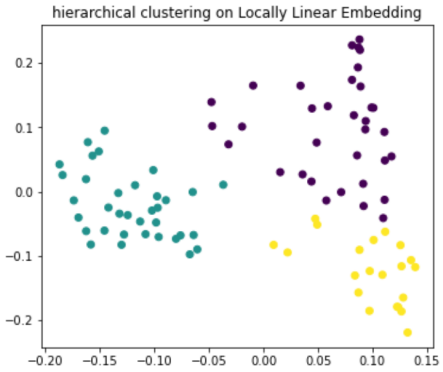


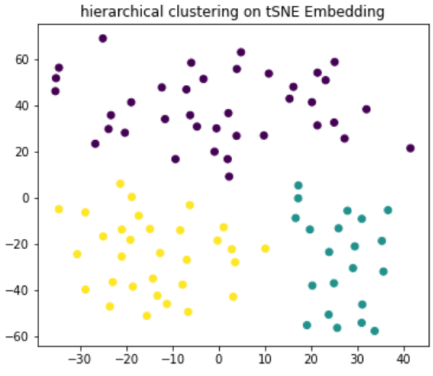


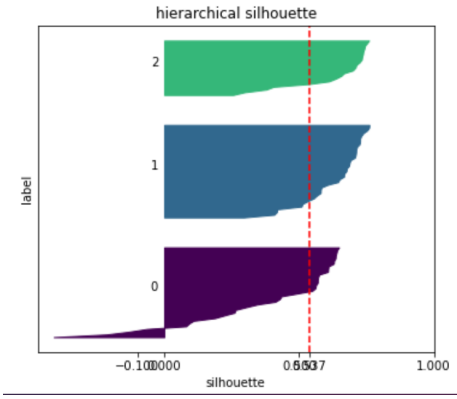


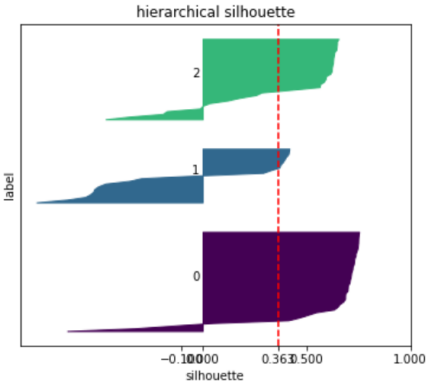


Feature Space 3 Clustering









References

[1] Thorsten Joachims. *Learning to classify text using support vector machines*. Vol. 668. Springer Science & Business Media, 2002.

[2] Thibault Langlois and Gonçalo Marques. "Automatic music genre classification using a hierarchical clustering and a language model approach". In: *2009 First International Conference on Advances in Multimedia*. IEEE. 2009, pp. 188–193.