Topological Data Analysis Methods Applied to Music Data and Genre Classification

Lily Yu
University of California, San Diego
alyu@ucsd.edu

Abstract

Feature extraction and engineering is an important step in the analysis of music audio data, as the raw audio data cannot easily be directly fed into statistical models. We believe that we should be able to find nontrivial topological features in music data in the spaces of time and pitch. We attempt to use Topological Data Analysis methods to extract features from music audio data, and use these features to classify music into genres.

1 Background

1.1 Music Classification

Music is difficult to summarize and represent visually or textually. Musical genres, though imprecise and subjective, convey information about music much faster than the low-bandwith medium of listening to an audio file, and enable individuals to search for categories of music that they are interested in. The subjective and fuzzy boundaries of genres make Music Genre Classification (MGR) difficult. It is an ongoing problem in the field of Music Information Retreival (MIR).

1.2 Audio Data

Music as it is distributed in the real world cannot easily be fed into classification models. The most commonly used audio formats are time series obtained by sampling from sound waveforms. The time series are generally "low density" - for better audio quality, more points must be sampled, but this means that any single point in the time series encodes little actual information that is meaningful to humans (Scaringella [2014]). For these reasons, effective feature extraction and/or engineering is a vital step in any music classification pipeline

1.3 Topological Data Analysis and Music

For the purposes of this report, we assume that the reader is familiar with Persistent Homology and other commonly used tools in Topological Data Analysis (TDA). Before we apply any TDA tools, we should show why we expect to find underlying topological structure in music data in the first place. To that end, we expect to see meaningful topology on the axis of time - in the musical concepts of repeating time such as *beats* and *measures* and more abstract ideas of structure through time such as *choruses* and *verses*, as well as the axis of pitch - in *pitch interval* and *harmony* relationships in music, and more abstract ideas like the *circle of fifths*.

1.3.1 Musical Time

In standard Western musical notation, music is divided into the basic time units of *beats* and *measures*. A *beat* is the most basic unit of time, and corresponds to a real number of seconds, often expressed as *beats per minute* (BPM). A *measure* is a larger unit of time and consists of multiple *beats*. We expect to see periodicity in music audio data on the scale of *beats* and *measures*, and hope to capture this periodicity using a delay embedding (Takens [1981]) and Topological Data Analysis methods.

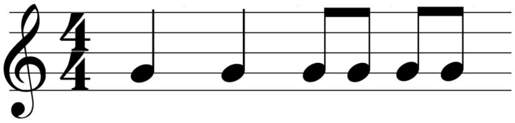

Figure 1: A *measure*, denoting four *beats*

Music is also often divided into much broader units of time denoting large sections of the music, such as *choruses* and *verses*. We hypothesize that *choruses*, *verses*, and other broad sections of music are generally distinct enough that we could see them appear as distinct connected components.

1.3.2 Pitch

In standard Western musical notation, pitch is expressed as a repeating 12-note scale, notated as A, A#, B, C, C#, D, D#, E, F, F#, G and G#, in ascending order of pitch. A thorough explanation of harmonic relationships between pitches is outside of the scope of this report, however, to put it very simply, pitches that are a particular distance apart tend to sound pleasant together. Western music theory captures these relationships with the idea of *pitch intervals*.

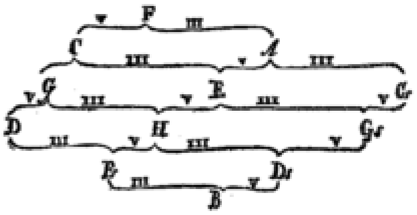

Figure 2: Euler's Tonnetz, representing notable musical intervals between pitches

The *Tonnetz* is a conceptual representation of pitch proposed by Euler that shows harmonic relationships (third and fifth intervals) common to European classical music (Euler [1739]). Some modern implementations of the Tonnetz map pitches onto geometric spaces, encoding interval relationships. Harmonic pitch intervals are a tool very commonly used by composers and musicians. Because of this, we expect to find nontrivial topological features in these geometric representations of pitch. The specific implementation of the Tonnetz in the `librosa` library that we use in this report encodes pitch information in a 7-dimensional space, capturing the interval relationships of *perfect fifths*, *major thirds*, and *minor thirds* (Harte et al. [2006]).

2 Methodology

2.1 Libraries Used

We used the Python package `librosa` (McFee et al. [2021]) to handle audio files and extract pitch features. We used `gudhi` (The GUDHI Project [2021]) to implement Topological Data Analysis methods. We used `scikit-learn` (Pedregosa et al. [2011]) to handle the features created using TDA and create classification models.

2.2 Dataset

We chose to use the Free Music Archive (FMA) dataset (Defferrard et al. [2017]), as it is a large corpus of audio files tagged with genres. It is worth noting that the providers of this dataset are able to share audio files because all of the music in the dataset is not copyrighted. This means that the dataset is likely not an accurate representation of the music most readers would be familiar with, as much of popular, widely-known music is copyrighted.

2.3 Time-Derived Features

We quickly discovered that if you attempt to load in an audio file, embed the waveform using Taken's embedding, construct a Rips Complex and attempt to compute the persistence using default parameters, you will run out of memory. Given our data, this makes sense. Most of the audio files in our dataset were sampled at 44 kHz - over 44000 observations for every second of music. This level of granularity is important for high-quality audio, but is too much for the purposes of our analysis.

To solve these problems, we downsampled our audio to 200 Hz (200 observations per second), and after embedding the waveform, we further sparsified our points such that all the remaining points had minimum squared distance above a threshold.

We also adjusted the Taken's embedding parameters. The implementation in `gudhi` has a default delay parameter of 1, which does not make sense for our particular application. Instead, we set the delay parameter to be equal to the sample rate, representing a delay of 1 second of audio.

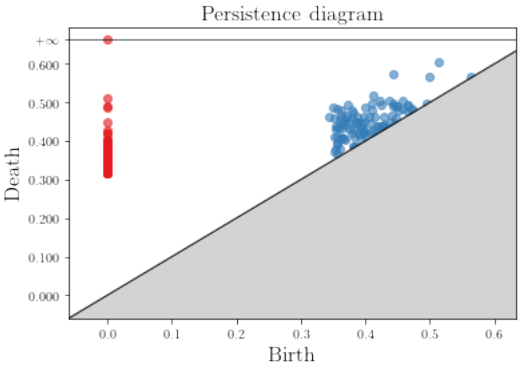

Figure 3: Persistence diagram computed from the embedded waveform of "AWOL - Food", the first track in our dataset

2.4 Pitch-Derived Features

`librosa` includes functions for extracting pitch features from audio files. These features are of varying dimensions. The Tonnetz, for example, expresses pitch-interval relationships in 6 dimensions, over a 7th dimension - time. Treating each section of a time as a data point, we constructed a Rips Complex over the Tonnetz, and computed the persistence. One example is displayed below.

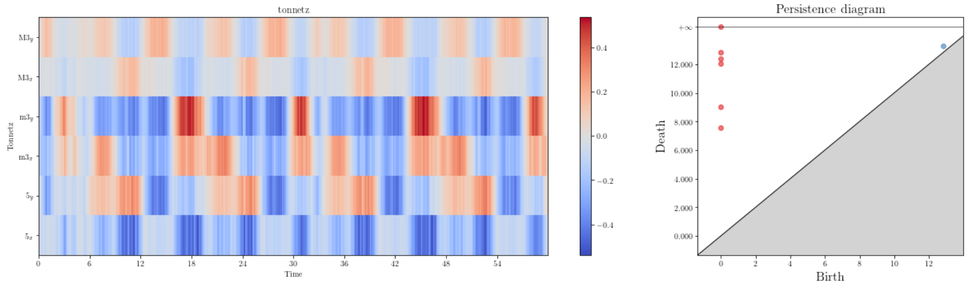

Figure 4: Persistence diagram computed from the Tonnetz of "AWOL - Food"

We also created persistence diagrams for all pitch features created by `librosa` that are 2 dimensional or higher. One example, the Mel Spectrogram, is shown below.

It should be noted here that the author lacks domain expertise in thoroughly understanding all of these pitch features. We leave a more thorough topological analysis of these features as future work.

2.5 Classification

We first attempted to use the Sliced Wasserstein Kernel to compute distances between persistence diagrams usable for classification (Carrière et al. [2017]). Using distance matrices computed on the persistence of raw waveforms as well as the Mel Spectrogram, we attempted to create Hierarchical Clustering Trees. However, after splitting the trees into 8 groups (the number of distinct genres in our dataset), we were unable to obtain a classification accuracy better than random chance for either feature used.

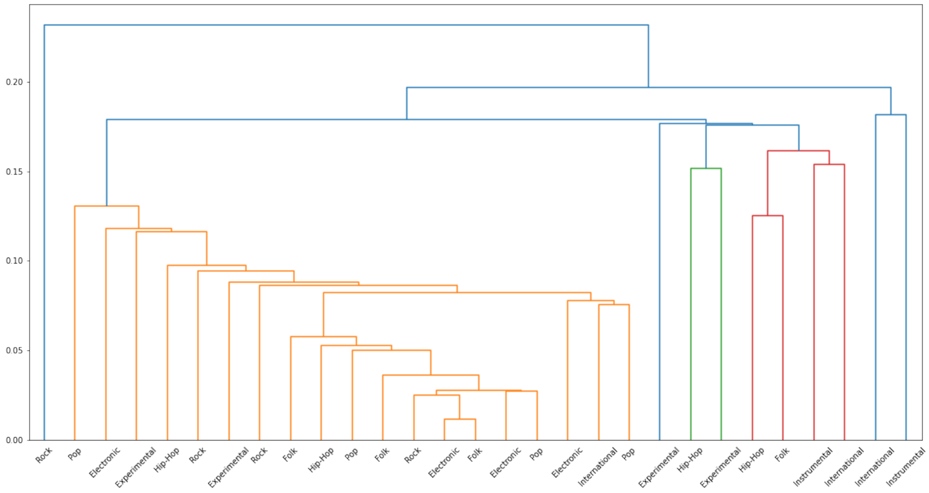

Figure 5: A Hierarchical Clustering Tree created on a small subset of the FMA dataset

We also tried using Persistence Images and Persistence Landscapes to represent our data for classification. We used a subset of our data, 800 tracks, with a 90/10 train test split. We then performed a 5-fold grid search to find the best combination of persistence representation out of [Persistence Image, Persistence Landscape, Sliced Wasserstein Kernel] and classifiers out of [Decision Tree, Random Forest, SVC]. We found that Persistence Landscapes with a Random Forest Classifier gave the highest accuracy for both our raw waveform persistence feature and our Mel Spectrogram persistence feature. Our accuracy was higher than random assignment of labels, but not higher than a baseline provided with the dataset (However, this baseline was trained on a much larger section of the data). We report our accuracy on the test set below.

Method	Accuracy
Random Assignment	12.5%
Raw Waveform Persistence	**19%**
Mel Spectrogram Persistence	**21%**
FMA Baseline	61%

3 Conclusion

We consider these results inconclusive - we were not able to achieve great accuracy scores in classifying music with our engineered TDA features, however, we were limited by time and computational capacity. We expect that if one were to run the same pipeline on a larger portion of the FMA dataset, they could achieve higher accuracy scores. Furthermore, individuals with greater domain expertise in signal processing and the handling of music data could likely create more meaningful features to be fed into the TDA pipeline.

References

F. Takens. Detecting strange attractors in turbulence. 1981.

L. Euler. Tentamen novae theoriae musicae. In *St. Petersburg: Imperial Academy of Sciences*, volume 1739, pages 1–263. 1739.

Christopher Harte, Mark Sandler, and Martin Gasser. Detecting harmonic change in musical audio. 10 2006. doi: 10.1145/1178723.1178727.

Brian McFee, Alexandros Metsai, Matt McVicar, Stefan Balke, Carl Thomé, Colin Raffel, Frank Zalkow, Ayoub Malek, Dana, Kyungyun Lee, Oriol Nieto, Dan Ellis, Jack Mason, Eric Battenberg, Scott Seyfarth, Ryuichi Yamamoto, viktorandreevichmorozov, Keunwoo Choi, Josh Moore, Rachel Bittner, Shunsuke Hidaka, Ziyao Wei, nullmightybofo, Darío Hereñú, Fabian-Robert Stöter, Pius Friesch, Adam Weiss, Matt Vollrath, Taewoon Kim, and Thassilo. librosa/librosa: 0.8.1rc2, May 2021. URL `https://doi.org/10.5281/zenodo.4792298`.

The GUDHI Project. *GUDHI User and Reference Manual*. GUDHI Editorial Board, 3.4.1 edition, 2021. URL `https://gudhi.inria.fr/doc/3.4.1/`.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

Michaël Defferrard, Kirell Benzi, Pierre Vandergheynst, and Xavier Bresson. FMA: A dataset for music analysis. In *18th International Society for Music Information Retrieval Conference (ISMIR)*, 2017. URL `https://arxiv.org/abs/1612.01840`.

Mathieu Carrière, Marco Cuturi, and Steve Oudot. Sliced wasserstein kernel for persistence diagrams. *CoRR*, abs/1706.03358, 2017. URL `http://arxiv.org/abs/1706.03358`.