CS 6170 - Project Progress Report

Yash Gangrade (u1143811) and Siddhant Ranade (u1141131) April 2, 2019

1 High-level Project Description

In this project, we are exploring the topological structures and feature in the Music data. Music in general is rich in structure. We are looking at similarity measures and distances between single notes, sequence of notes (ordered), beats etc. and we are looking at the topological structures present in these. We extract these topological structures and features for different types of music data and we want to apply them for getting results of the higher classification tasks like genre classification, composer identification, comparison of chord progression and so on.

2 List of completed milestones

- Data collection pipeline, and significant progress on data collection. It involves downloading the MIDI file and selecting an appropriate channel (i.e. instrument) for that track. Each track takes roughly a minute to process.
- Point-cloud extraction: Given a MIDI file, there are several ways to extract point-clouds for persistent homology. These include a simple single note extraction (i.e. extract the set of all the notes in the song), time-series embedding (i.e. extract the set of all note sequences of length N), chord-class embedding (i.e. extract the set of all chords in the track), chord sequence embedding (i.e. extract the set of all sequences of chords of length N). Of these, we have finished the single note embedding and time-series embedding. We have also made some progress on chord extraction (chord sequence embedding naturally follows from this).
- Given a point-cloud, we compute the distance matrix for the specific class of point-cloud being used, as input to ripser for computation of persistence diagrams/barcodes. We have code to do distance computations for the single note embedding and the time-series embedding.
- Then, we apply distance measures like Bottleneck or Wasserstein distances to compute the dissimilarity between these diagrams. We plot an embedding of these distances using TSNE (just like HW2).

3 List of upcoming milestones

- Finishing data collection.
- Finishing chord-class extraction and chord-sequence embedding.
- Distance measure for the aforementioned embeddings.
- Use machine learning algorithms along with neural networks for the artist and genre classification based on the songs. We will compare it to the state of the art as discussed in Catalteple et. al.
- Identifying the common chord progressions in popular music (e.g. Imagine Dragons, Coldplay, and similar). It basically uses 4 chords transposed to whatever key the song is in – can we identify these? We do want to compare it with classical music such as Beethoven's piano sonatas, for instance.

4 Preliminary results

The results and procedure for the current milestones is discussed in different sections as follows.

4.1 Data Collection and Processing

The data collection for us has been more of manual labor where we find the data and extract channels from it manually.

- Download the midi files for different type of songs, for e.g., pop, rock, classical etc. from any of the online source like Bitmidi. We have used songs from Queen, Coldplay etc. bands along with solo artist songs such as Taylor Swift, as well as we have used classical sonatas like Beethoven, Fur elise etc. Having the diversity in the data provides us new insights and also help us to prove the correctness of the project. Now we have the raw data for our project.

- Use a open-source software called MuseScore to manually select a channel from the midi files and then exporting it as another midi file. Channel selection is based on the information each channel provides, usually for topological data analysis, piano, drums, guitars are considered good channels because they have some patterns we can find. At this point, we are selecting one channel per midi file but for the future milestones, we will use multiple channels for a midi file. Please find the attached screenshot of how the channel selection works in the software.

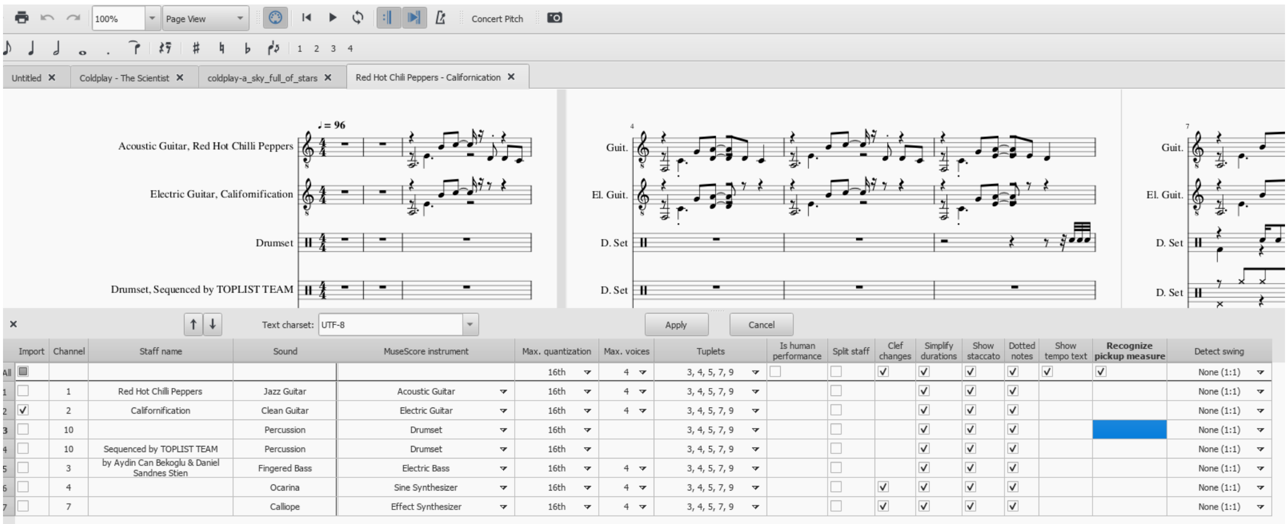

Figure 1: Channel Selection in MuseScore

4.2 Extracting point cloud and analyzing persistence diagrams from the data

- Read the file using mido library in python. Then we have implemented some helper functions to create point cloud from the extracted data. One of the functions we have implemented makes a list of lists in python where each element is the timestamp in the midi file along with the midi message. Midi message is just the information about which note was pressed on which channel with what velocity (eg. pressure applied on piano key) for what amount of time. Then, finally we implemented a function which preserves the time series while creating the point cloud. Essentially we follow takens embedding to create this point cloud. We decide the number of dimensions we want in the point cloud, for eg. with 3 we will get *(note1, note2, note3); (note2, note3, note4)* and so on.

- Once we have the point cloud, we use another helper function to compute the distances between the points in the point cloud. Here, we also use the fact that the notes are cyclic at every 12 notes difference. Then we have a distances matrix which we pass along to the ripser to compute the persistence diagrams.
- We perform the aforementioned procedure for all the midi files and store all the persistence diagram in form of a list. Once we have this data, we compute all the bottleneck and wassertian distances for all pair of persistence diagrams. This part is similar to what we did for project 2. Finally, we plot the distances after applying the dimensionality reduction techniques specifically TSNE because it works well in preserving the distances.
- Please find the results in form of the TSNE plots over Wasserstein and Bottleneck Distances in the subsequent page. As of right now, we are using 30 midi files to show our work but we plan to use about 100-120 midi files to achieve the rest of the milestones. These 30 midi files have 3 different artists namely Queen (10 songs), Coldplay (15 songs), and Red Hot Chilli Peppers (5 songs).

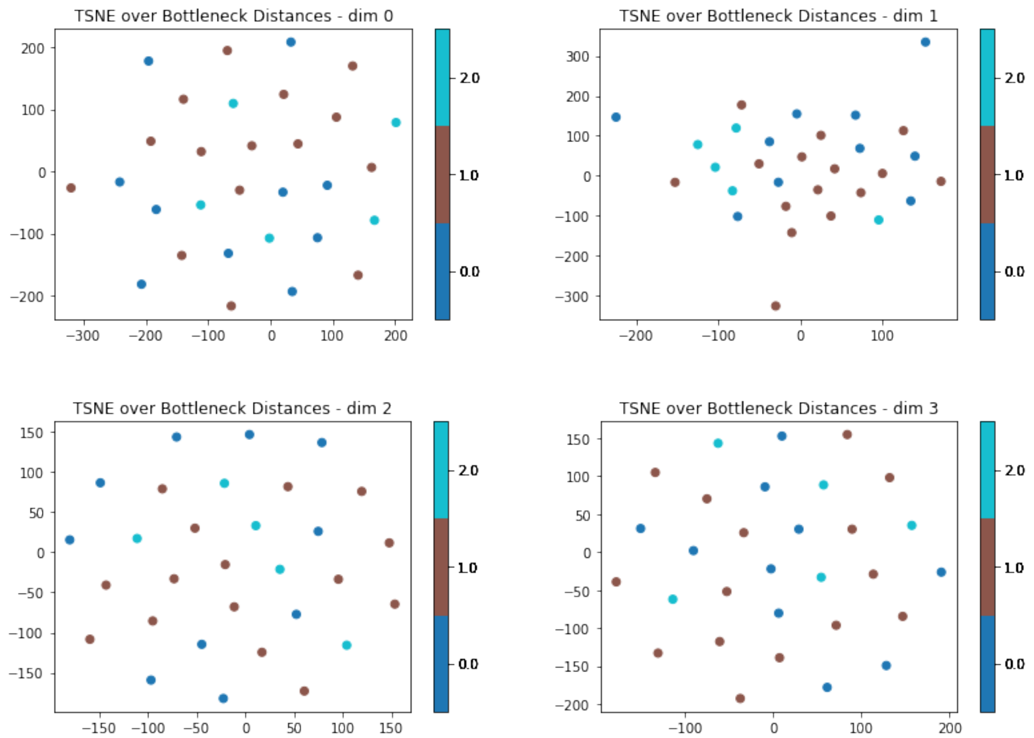

Figure 2: TSNE applied to Bottleneck distances over persistence diagrams. Here, the number of files = 30 with 3 different artists and the persistence diagrams are computed up to dimension 3. Clearly, we can see that the clustering is not so strong in any one class as a whole because all the songs that we have are different from each other. But we can see that some songs within the class and when compared to the other classes are similar to each other, e.g. in dimension 1, artist 2 and artist 0 have similar homology

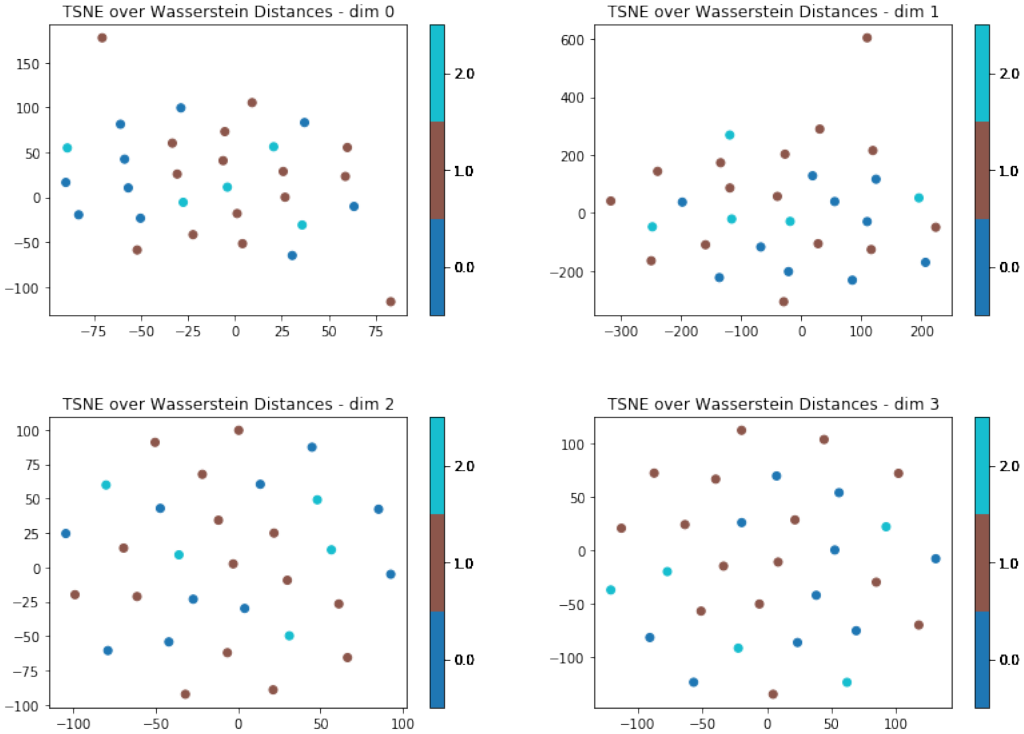

Figure 3: TSNE applied to Wasserstein distances over persistence diagrams. Here also, we can see that the clustering is not so strong in any one class as a whole because all the songs that we have are different from each other. Some songs are completely different than others, for eg. in dimension 0, two songs of artist 1 are very different from all other songs. But we can see that some songs within the class and when compared to the other classes are similar to each other, e.g. in dimension 2, artist 1 and artist 0 have similar homology

5 Updated Milestones, Any modification, and Project on Track?

At this point, we are not adding or removing any of our milestone. We believe that we are on track to complete our project by the end of the month. During these initial three weeks, we mostly spent our time learning about the structures in the music data as well as collecting and processing the data. We also tried a few basic tests to see if the data is reasonable or not.

6 Summary

Summary of our project is exploring and analyzing interesting topological structures and answer several real-world questions for the music data. Some of the questions we would like to address are the relation between chord progressions, genre classification, artist classification etc. We are using a diverse dataset so as to make it as close to the real world as possible.