Categorical Prediction of Song Popularity Using Topological Data Analysis

Michael Lin[2], Callie Mao[1,2], Grace Mao[1], Kelvin Niu[1,2], Kevin Tie[1]
[1]Department of Computer Science, Duke University, Durham, NC
[2]Department of Statistical Science, Duke University, Durham, NC

Introduction

What makes a song popular? With catchy new singles released every week, it is interesting to consider whether or not it is possible to predict song popularity based upon just its raw audio file. Is it possible to determine how much a song will be overplayed on the radio based on characteristics of its audio amplitudes and frequencies? Are there some qualities that number 1 smash hit songs possess that perhaps songs floating around the rank 80's do not? Do the best songs share a particular type of rhythm or style that makes them stand out from the vast sea of songs that are more forgettable to the general audience?

For this project, variables that could objectively measure a song's flow, beat, tempo, tune, etc. were evaluated from parsed audio files of songs of different ranks on the monthly Billboard 100 list. The purpose of this project was to focus on finding a musical pattern that could influence how high a song rose on the charts.

Ideally, many of these parameters would be statistically significant in predicting the popularity of a song, suggesting that the model would be able to accurately predict the popularity of new songs before they are released. However, there is also the possibility that no noticeable difference is described by the chosen parameters, in which case the search for the magic ingredients will have to continue in a later project.

Data Collection

Top songs ranked 1, 21, 41, 61 and 81 from Billboard 100 over the last 13 months were chosen as the data set of songs for analysis. The songs were specifically chosen to be 20 ranks apart to maximize possible differences for classification.

Each audio file, obtained from the web, was in MP3 format and was imported into MATLAB as an n x 2 matrix, where n is the number of samples. The entries in the matrix are signed values between -1 and 1, and each column of the matrix represents one of two channels. The matrix represents the amplitude of the sound wave over time, normalized to between -1 and 1. For simplicity, a vector of only the first column (left channel) was used for analysis.

To compute the frequencies at each time point, the amplitude matrix was transformed using the "Spectogram" function in MATLAB, which calculated the discrete Fourier transform on each subset of the data. The magnitudes of these transformed values were then obtained and returned as a vector of frequencies.

For computational tractability, both the vectors for amplitude and frequency were compressed by only retaining the maximum magnitude value for each 0.1 second of audio. As a result, each vector should have 10 times as many data points as number of seconds in the song.

Amplitude and Frequency

Below are sample amplitude and frequency plots to visualize the data used for analysis:

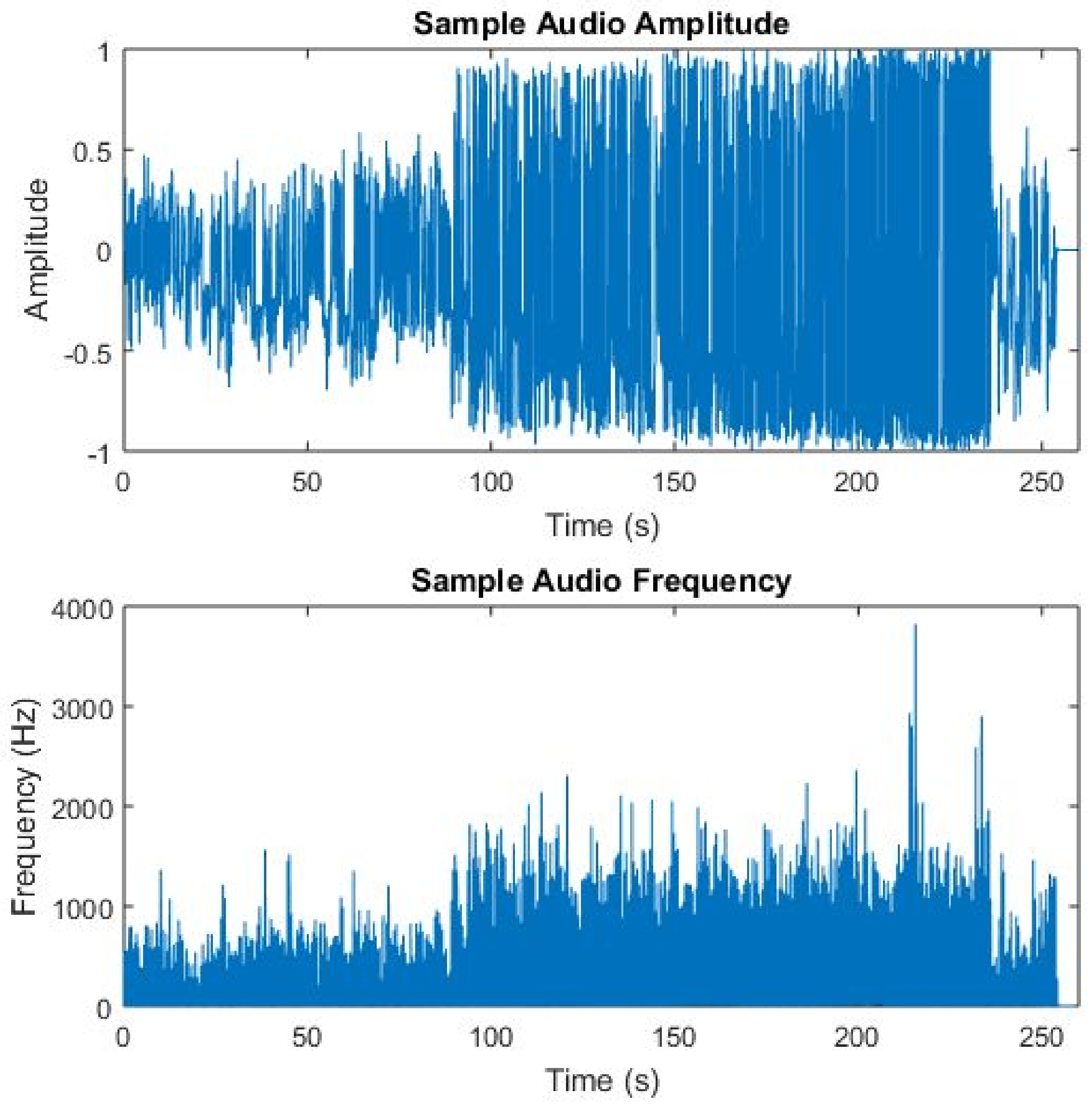

Figure 1
Sample amplitude and frequent plots

Parameter Selection

To capture the time progression of songs, each song was broken down into 3 second frames, which were each characterized with certain parameters. Consecutive frames were overlapped by 1 second in order to retain connectivity and a sense of the song's flow. The following parameters were selected for calculation to potentially capture qualities of the songs:

- Sum of frequencies - summary of how high frequencies are
- Sum of absolute values of intensities - summary of how high intensities are (how loud the song is)
- Sum of absolute values of differences between consecutive frequencies - summary of frequency fluctuation
- Sum of absolute values of absolute differences between consecutive intensities - summary of intensity fluctuation
- Number of frequencies greater than one standard deviation of the total mean - summary of high-frequency outliers
- Percentage of amplitudes with absolute values between 0 and 0.5 - summary of small amplitudes

Thus, each frame was represented as a point in 6-D space and each song was represented as a point cloud. Finally, all parameters were normalized to the global maximum (over all frames of all songs) to shrink all point clouds into a unit cube and put equal weight on each of the parameters.

Data Analysis

Each song represented as a point cloud in 6-D space was further processed to produce the following characteristic values per song:

- Sum the values of each parameter to produce six distinct values
- Analyze the data via Topological Data Analysis (TDA) tools and find
  - The total number of 1-cycles per song
  - The sum of all persistencies of the 1-cycles

In geometry, a k-simplex is a k-dimensional polytope that is the convex hull of its k+1 vertices.[1] In algebraic topology, a k-chain is a linear combination of k-simplices.[2] Furthermore, the boundary of a k-chain is a linear combination of the boundaries of the simplices in the chain, and a k-chain is a k-cycle if its boundary is zero. A 1-cycle is thus a 1-dimensional boundary of any 2-dimensional object.

In the TDA, the radius of each point in the point cloud was expanded over time as the birth and death of 1-cycles were tracked. A 1-cycle is born when the radii of nearby points overlap and merge to form a 1-dimensional closed loop; it dies when the radii expand to fill the hole in the loop. The persistence of a 1-cycle is the difference in time between its birth and death. Analyzing 1-cycles and their persistences provides insight into the arrangement of the data points in space.

Shown below are the point cloud representations of three songs ranked 1, 41 and 81 from February 2015 considering only the first three parameters. Somewhat selective clustering of the rank 1 song (red) and rank 81 song (blue) can be seen.

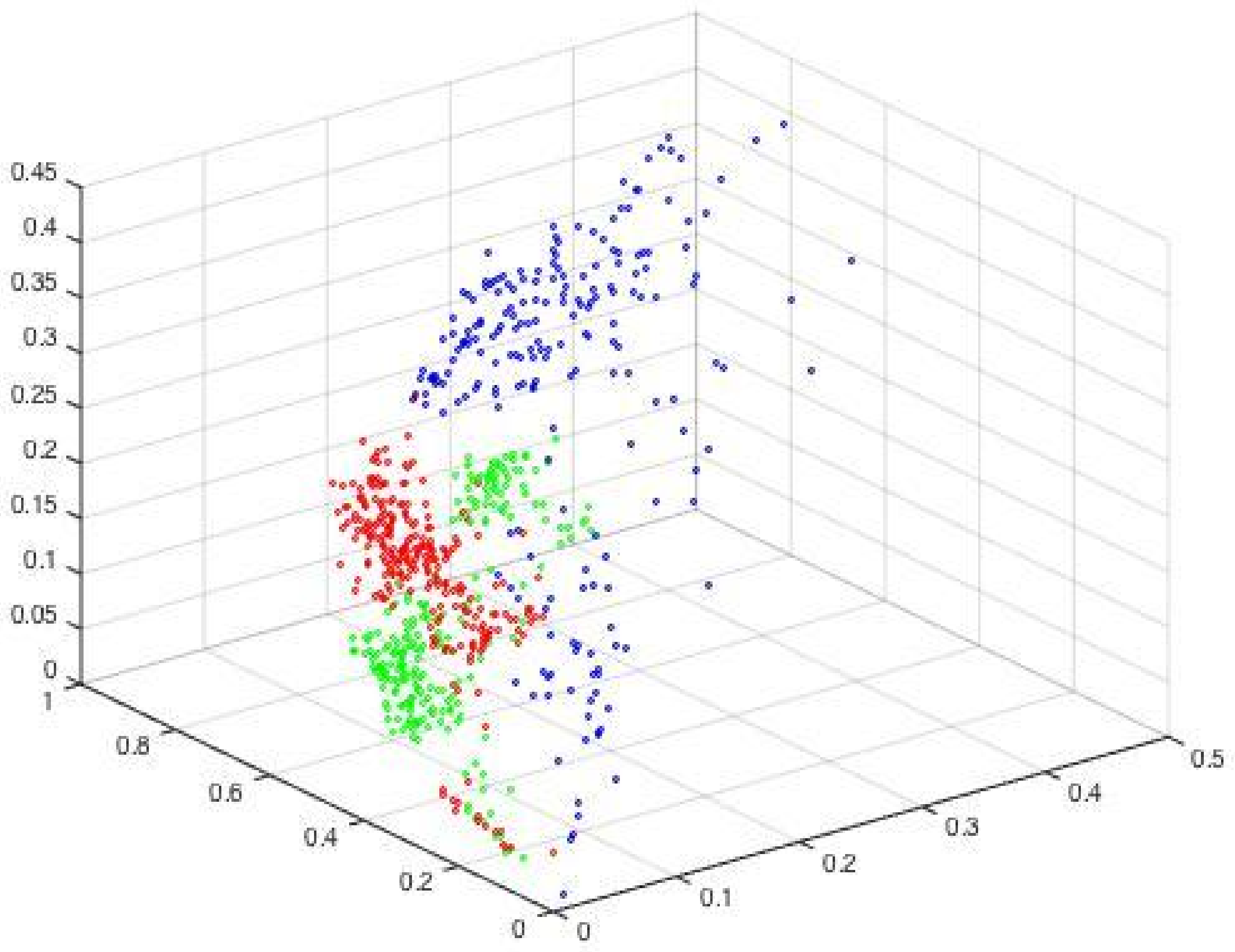

Figure 2
Point cloud representation of the first 3 dimensions from February 2015
(Red = rank 1, green = rank 41, blue = rank 81)

Permutation Test

Hypothesis tests using permutation (perm) test as test statistics were performed on results from topological data analysis. In a perm test, data from two sets are randomly relabeled n times, and the proportion of times that the difference between the means of the newly labeled sets exceeds the original difference of the means serves as a p-value to determine the significance of the distinction between the two sets.[3] In our analysis, each pair of group rankings (e.g. rank 1 group vs rank 41 group) were chosen as set pairs. P-values were computed with n = 100,000 random group labelings for each of the "characteristics" mentioned in the bullet points under "Data Analysis." In particular, the perm tests yielded the following significant results at the 0.05-α level:

- Frequency fluctuations for (21, 61) and (21, 81) ranking group pairs
- Frequency outliers for (61, 81) ranking group pair
- Amplitude magnitude for (1, 61), (1, 81), and (41, 81) ranking group pairs

Results and Conclusion

- Considering only TDA heuristics, a significant difference was found between the following pairs of ranks: 1 and 41, 21 and 41
  - A somewhat significant difference between the following pairs of ranks: 1/81, 41/61, 61/81
- Considering all heuristics used in the pair-wise permutation tests, the following shows order of decreasing significance among pairs:
  - 1/61, 21/41, 41/61, 21/61, 1/81, 21/81, 41/81, 1/41, 1/21, 61/81
- Ignoring parameters 1 and 2 for perm tests:
  - 41/61, 1/61, 1/81, 21/41, 21/61, 41/81, 21/81, 1/41, 61/81, 1/21
- Roughly speaking, there seems to be a moderate difference when comparing high ranks (1, 21) to mid ranks (41), mid ranks to low ranks (61, 81), but no difference when comparing high ranks to low ranks
- The lack of decisive results could be attributed to (1) rudimentary parameters chosen for modeling, (2) insufficient data sample size, or (3) analysis methodology that is unable to fully capture all information in the data

Future Directions

For future extended analysis, it would be beneficial to increase the sample size and evaluate more data from all listed ranks for longer time periods. Instead of running only pairwise permutations, running TDA on all groups simultaneously would allow for better holistic results. Additionally, more detailed parameters could be analyzed to build a more comprehensive model.

References

1. http://mathworld.wolfram.com/Simplex.html
2. http://mathworld.wolfram.com/topics/AlgebraicTopology.html
3. Legendre, P. & L. Legendre. 1998. Numerical ecology, 2nd English edition. Elsevier Science BV, Amsterdam