DM Project 5

Alex Brockman, Andrew Haisfield, Anish Prasanna, Carlos Samaniego

November 8, 2019

1 Introduction

In this assignment, we were able to extract features from a .wav dataset using the LibROSA package. This dataset included common background noises, such as dog barks, car horns, and cat meows. We sought to cluster these .wav files into meaningful groups for further analysis. As part of our analysis methodology, we clustered these features using our own KMeans Algorithm, a Sci-Kit implementation of the KMeans algorithm, a Sci-Kit implementation of DBSCAN, and a Sci-Kit Agglomerative Clustering implementation. Additionally, We sought to identify files that were grouped in the same clusters, amongst these various clustering algorithms.

We began our assignment by extracting the features into a csv. From here, we ran DBSCAN to determine the optimal number of clusters, while also removing outliers. Next, we output the cluster membership of each file in a separate file denoting the algorithm used. Lastly, we built a membership chart identifying the number of instances files were grouped in the same cluster.

2 Finding Optimal EPS Value

In order to find the optimal EPS value with our data set, we ran code to output a plot that would indicate when the EPS value starts to rapidly increase. The plot is displayed in Figure 1 in our conclusion. We know that as soon as the EPS value starts to have a high slope our optimal EPS value has been reached. For our dataset we determined that the optimal EPS value was around 68.

3 Run Times

After testing all of our methods with a subset of files, we ran our CSV generation algorithm on all the data files. Using all 40 of the features, the output file took 32 minutes and 43 seconds to create. Once the CSV was created, we calculated the run times for DBSCAN, our K-Means, Sci-Kit Agglomerative Clustering, and Scit-Kit K-Mean. The results are as follows:

- DBSCAN: 0.035 seconds

- Our K-Means: 33.96 seconds
- Sci-Kit Agglomerative Clustering: 0.041 seconds
- Scit-Kit K-Means: 0.072 seconds

Interestingly, our K-Means function took the longest. Most likely this occurrence was due to our algorithm being less efficient than Sci-Kit's algorithms.

4 Concordance Analysis

After identifying the cluster membership between our KMeans, and the other Sci-Kit Clustering Algorithms, we build a Concordance Figure (Figure 2), depicting instances of files being clustered in multiple algorithms. As seen by the subsections of dark red, numerous files were found to be clustered in the same group as our KMeans implementation.However, a significant proportion of the files were clustered differently, as represented by the yellow, and orange portions of the Concordance Figure.

5 Conclusion

To surmise, we found an optimal EPS value to include in our "off-the-counter" DBSCAN algorithm by analysis, determined the run times for all of our algorithms and analyzed our cluster membership by building a concordance figure. We discovered that the DBSCAN algorithm was the fastest out of all of our options. While all these algorithms had varying efficacy in clustering, when working with even larger datasets, DBSCAN could save a significant amount of time.

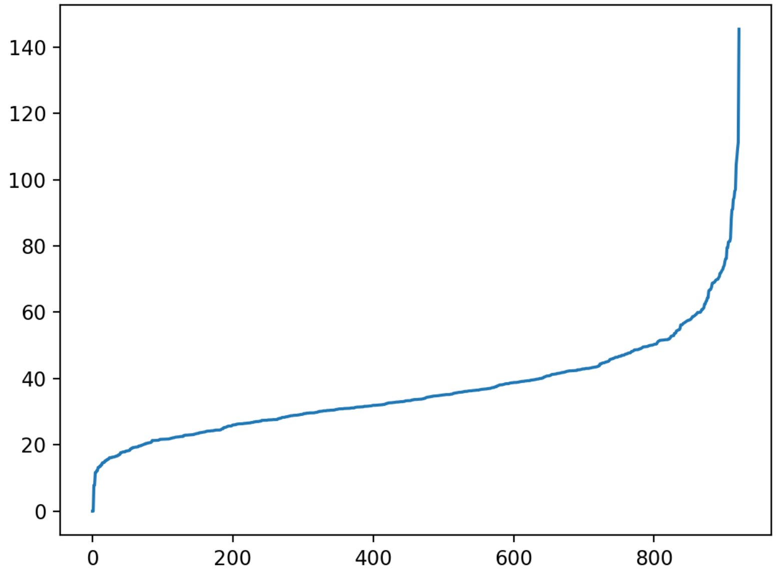

Figure 1: EPS Optimization Graph (X-axis represents the number of data points, Y-axis represents EPS value)

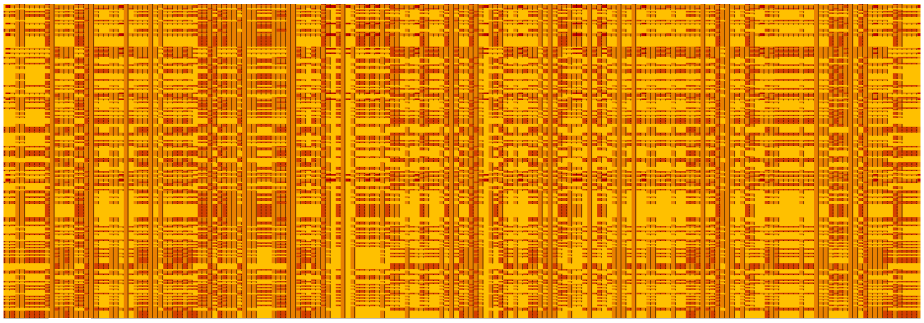

Figure 2: Concordance Table (Darker color denotes higher cluster membership amongst Clustering Algorithms)