Data Analysis Final Project

For this project, the goal was to use data analysis techniques discussed in class to conduct an open ended project in data analysis.

Background

Vocal range in operatic singing falls along six basic categories: soprano, mezzo-soprano and contralto for women and tenor, baritone and bass for men. While further divisions are recognized by different classification systems, this project will only deal with these basic categories. These vocal ranges follow an overlapping ordered scale. Bass corresponds to frequencies (approximately) of $80 - 330\ Hz$ (E_2 to E_4) in scientific pitch notation; baritone corresponds to $100 - 350\ Hz$ ($G_2\ to\ F_4$), tenor corresponds to $120 - 440\ Hz$ ($B_2\ to\ A_4$). On the other hand, contralto corresponds to $170 - 650\ Hz$ ($F_3\ to\ E_5$), mezzo-soprano corresponds to $220 - 880\ Hz$ ($A_3 - A_5$) and soprano corresponds to $260 - 1000\ Hz$ or higher ($C_4 - C_6$) [2].

While this system was made exclusively for classical singing, it can be applied to non-classical music as well. However, one should be wary that non-classical singers don't always follow operatic conventions.

For this project, the goal is to group the songs by the vocal range of the singer. However, in this case, since each vocal range has only one singer, this will be equivalent to grouping the songs by their singer as well.

Data

The data collected here is in the form of 60 songs from six different singers. Each singer has ten songs. The singers, in order, are: Kate Bush (soprano), Rihanna (mezzo-soprano), Adele (contralto), Michael Jackson (tenor), Elvis Presley (baritone) and Barry White (bass).

The particular songs used are:

Wuthering Heights, Cloudbusting, The Man with the Child in His Eyes, Breathing, Wow, Hounds of Love, Running Up That Hill, Army Dreamers, Sat in Your Lap, and *Experiment IV* by Kate Bush.

Kiss It Better, Desperado, Woo, Needed Me, Yeah I Said It, Same Ol' Mistakes, Never Ending, Love On the Brain, Higher and *Close To You* by Rihanna.

Hello, Send My Love(To Your New Lover), When We Were Young, Remedy, Water Under The Bridge, River Lea, Love in the Dark, Million Years Ago, All I Ask and *Sweetest Devotion* by Adele.

Don't Stop 'til You Get Enough, Rock With You, Workin' Day and Night, Get on the Floor, Off the Wall, Girlfriend, She's Out of My Life, I Can't Help It, It's the Falling In Love, Burn and *This Disco Out* by Michael Jackson.

A Big Hunk o' Love, I've Got a Thing About You Baby, Suspicious Minds, Don't, I Just Can't Help Believin', Just Pretend, Love Letters, Amazing Grace, Starting Today and *Kentucky Rain* by Elvis Presley.

and

You're The First,The Last,My Everything, You See The Trouble With Me, Can't Get Enough Of Your Love, Lady, Sweet Lady, Sheet Music, It Ain't Love Babe(Until You Give It Up), Let Me In And Lets Begin With You, We Can't Let Go Of Love, Your Love,Your Love and *Free* by Barry White.

Each song was originally encoded in the FLAC (Free Lossless Audio Codec) [1]. FLAC is a lossless audio codec, allowing full fidelity to the singer's voice. The sampling rate used was 44.1 kHz or 44,100 samples every second, with a bit depth of 16 bits per sample. This gives about 2,646,001 samples in a 60 second period. These samples are then used as features, giving us a dimensional space of $\mathbf{R^{2,646,001}}$. Each song is clipped to one minute starting from 45 seconds, the reason being that many songs begin with long silences with instrumental music only while we are interested in the voice of the singer.

Analysis

Initial Analysis

The most obvious choice for this task was k-means Clustering using LLoyd's Algorithm. However, there is one big issue that crops up here. The complexity of k-means. It is $O(nkdi)$, where n is the number of observations, k is the number of clusters, d is the number of dimensions in the data and i is the number of iterations. As can be expected, the running time for this problem won't be practical due to the high value of d.

Visualizing the Problem

The following are plots of the audio signal for one of the songs in the time domain and the average audio signal for all the songs, also in the time domain.

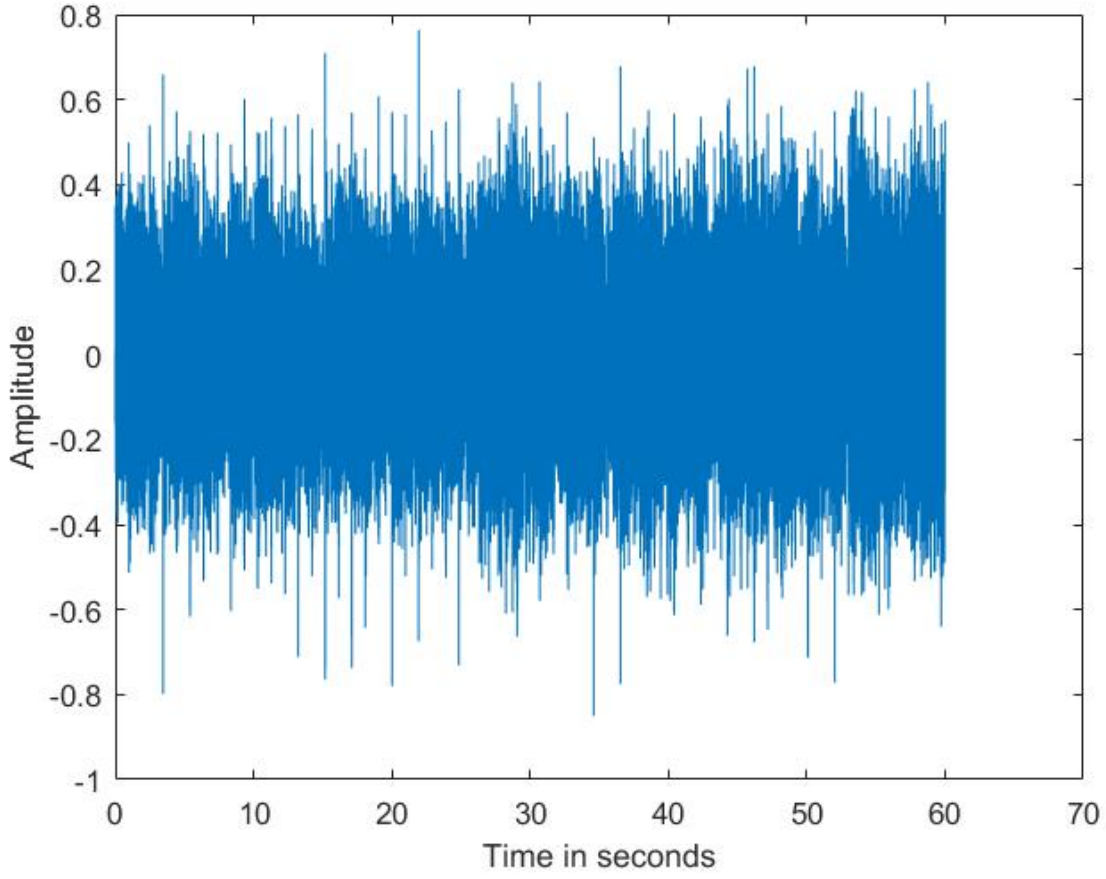

Figure 1: The time signal of *Wuthering Heights* by Kate Bush. The amplitude has been normalized to values in [-1,1].

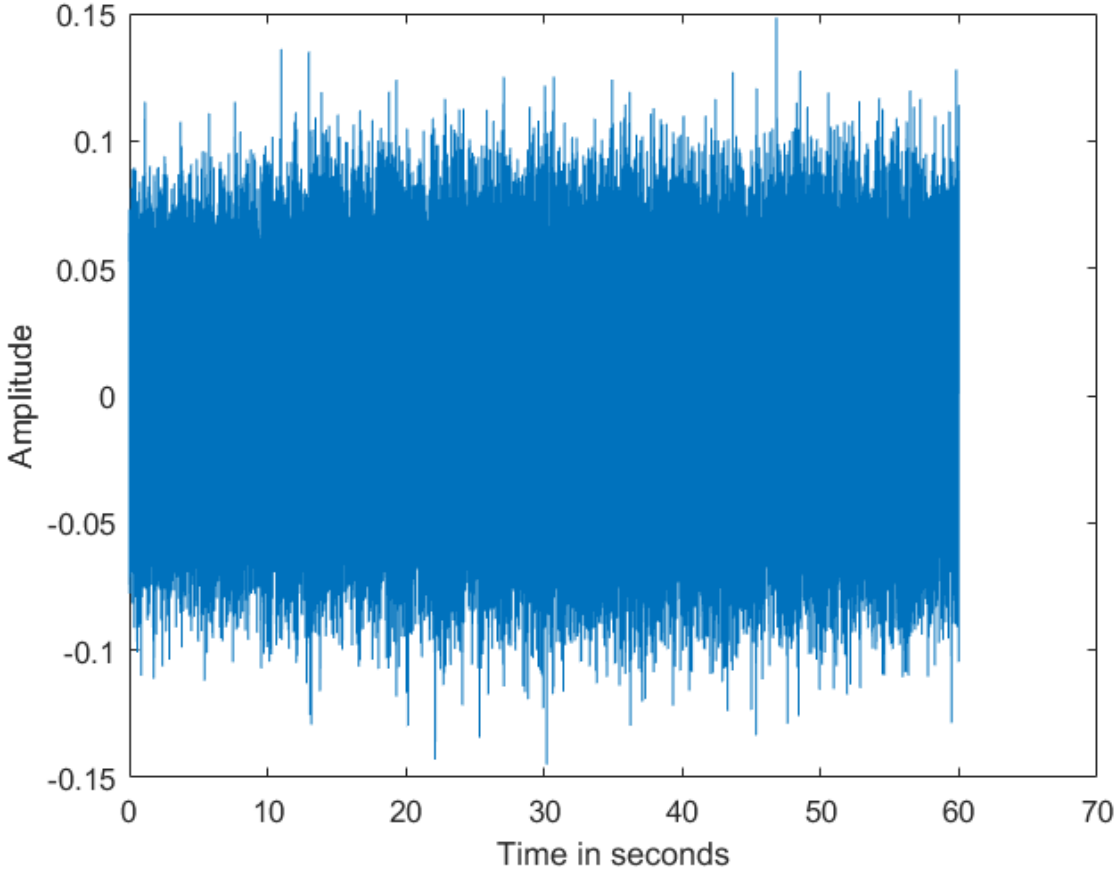

Figure 2: The average time signal of all the songs. The amplitude has been normalized to values in [-1,1].

PCA

One can look towards PCA as the savior of the day due to it's linear dimensionality reduction features which should fit in well with the linearity of audio samples. However,

```
[coeff,~,latent] = pca(songs');
```

produces a 59x1 vector because this is an underdetermined system. Moreover, the amount of variance explained by latent is only $\approx 50\%$ of the variance in the data, making these principle components insufficient.

Fourier Transform

Another technique that yielded unsatisfactory results was the Short-time Fourier Transform. The data was converted to spectrograms using the Short-time Fourier Transform. This variant of the Fourier Transform maps the signal's frequency domain in multiple time windows so as to preserve some of the information about the time domain. However, this didn't work as well. After the signal

was converted to the frequency domain, taking it's inverse didn't preserve the original information and the resulting signal was not the same as the original.

Johnson-Lindenstrauss

The Johnson-Lindenstrauss Lemma [3] provides for an easy way to reduce dimensionality as long as the goal is to minimize error between pairwise distances between the original data and the reduced dimensionality data. If k, the reduced dimension, is set to $1/\epsilon^2$, we get error ϵ. We set our error to five percent, giving us $k = 400$. Following is the code used:

```
%% regular jl
k = 400;
X = songs;
error_jl = zeros(length(k),1);
P = randn(k,size(X,1))/(sqrt(k));
y = P*X;
dist_jl = zeros(size(X,2));
% calculate pairwise distances for jl reduced dim data
dist_jl = squareform(pdist(y'));
% find error for jl
error_jl = max(max(abs((dist_jl.^2)./(dist_orig.^2)-1)));
avg_error = abs((dist_jl.^2)./(dist_orig.^2)-1);
avg_error(isnan(avg_error))=0;
avg_error = mean(mean(avg_error));
```

The resulting matrix has dimesnions of 400x60, far more manageable than 2,646,001x60. The average error is 5.41 %.

Clustering

After the dimensional reduction, the data is ready to be clustered. The following code performs the clustering:

```
%% Clustering
original_idx = [repmat(1,10,1)' repmat(2,10,1)' repmat(3,10,1)' repmat
    (4,10,1)' repmat(5,10,1)' repmat(6,10,1)'];
idx = kmeans(y',6); % make 6 clusters
error_k_means = sum(idx~=original_idx')/60
```

where y is the reduced dimension matrix of songs.
The resultant error is dismal: 83.33 %.
To improve upon this, a diffusion map was used on the reduced dimension data. The weight matrix was composed using the epsilon-neighborhood technique since it was easier to implement than the Gaussian distance. The resultant error was still high: 79 % for time step $t = 1$. Further iterations of t did not improve upon this.

The following silhouette plot futher demonstrates the bad clustering:

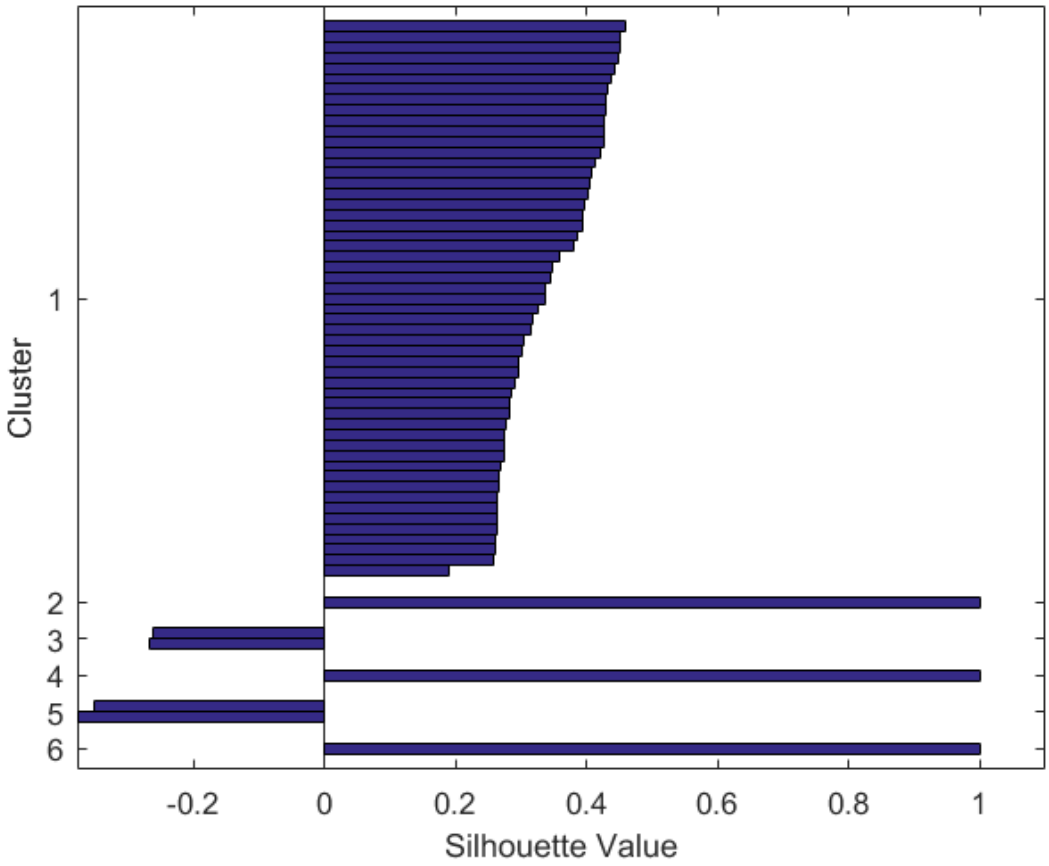

Figure 3: Silhoette plot of clustering done with diffusion maps.

One of the problems that k-means faces is that it's not robust when the clusters are not separated linearly. However, spectral clustering and diffusion maps reduce this weakness by using a similarity matrix, given by $A = D^{-1}W$. This similarity matrix depends upon W, the weight matrix. In this case, ϵ was chosen to be 590, a value that makes W almost useless. The reason for this was that otherwise D had some zeros in its diagonal, making it impossible to find its inverse. This led to a bad W, leading to the less than robust analysis by diffusion maps.

If there was more time and computation power, the clustering can be improved much more by using the Gaussian kernel function as the distance function in calculating W and by keeping a higher number of feature in the feature reduction step.

Another way to improve the clustering would be to include more observations for each of the vocal ranges.

References

[1] FLAC. https://xiph.org/flac/.

[2] International Pitch Notation. http://www.flutopedia.com/octave_notation.htm.

[3] Johnson-Lindenstrauss. http://www.cs.ubc.ca/~nickhar/W12/Lecture7Notes.pdf.