ENGR-E 511 Fall 2018: Assignment #2

Due on Friday, September 30, 11:59P

Professor Minje Kim

Abhilash Kuhikar (akuhikar@iu.edu)

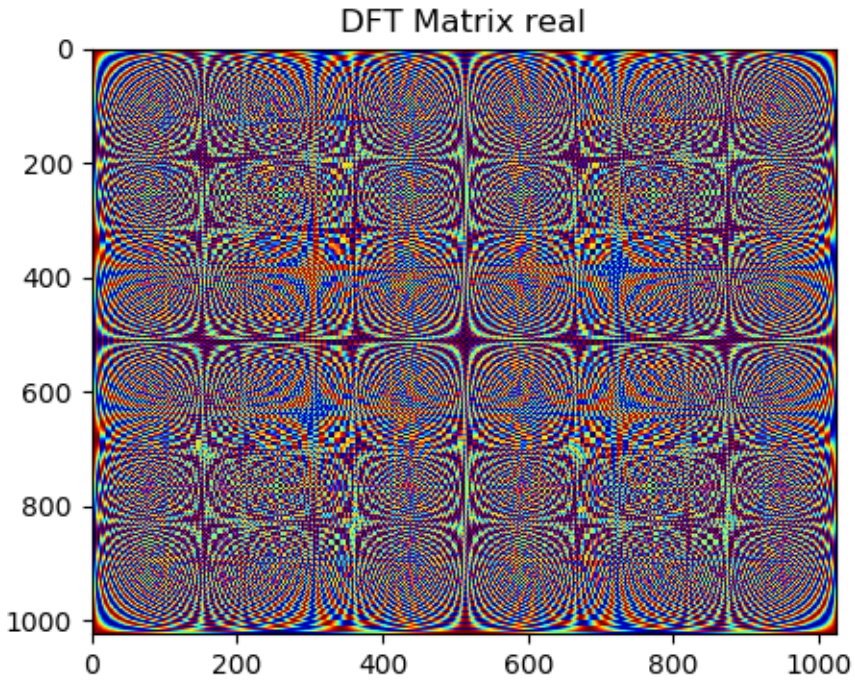
September 30, 2018

Contents

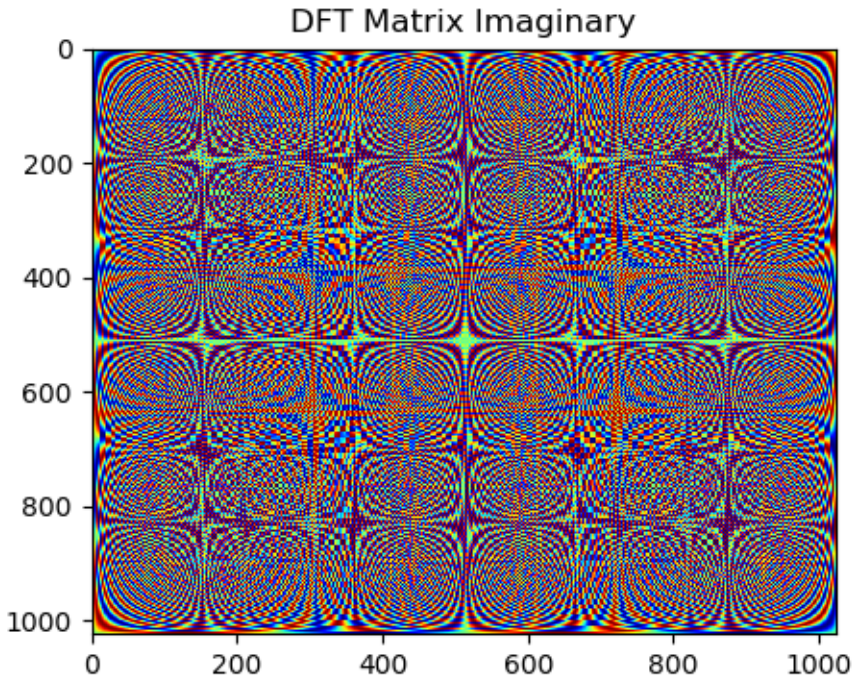
Problem 1: White Noise

The objective is to use STFT on the input sound signal to reduce or suppress the noise in the data

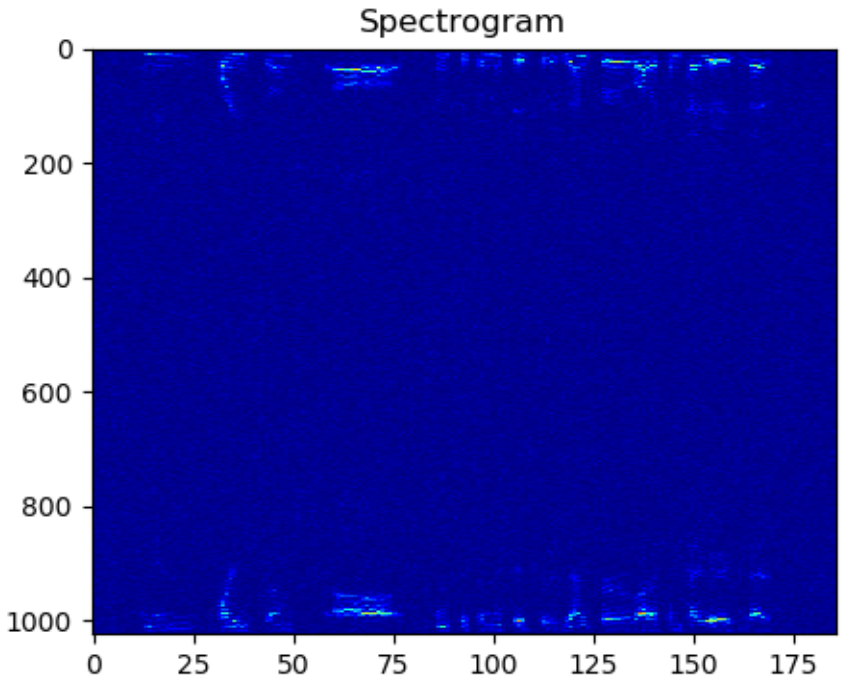
(a) The first step is to create a DFT matrix for Fourier Transform:



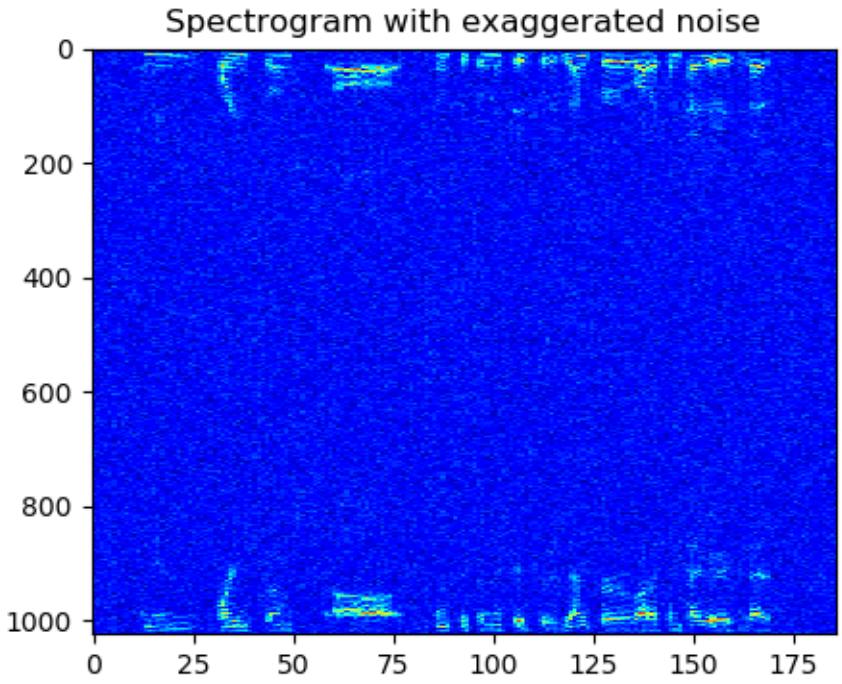
(b) The imaginary part of the DFT matrix:



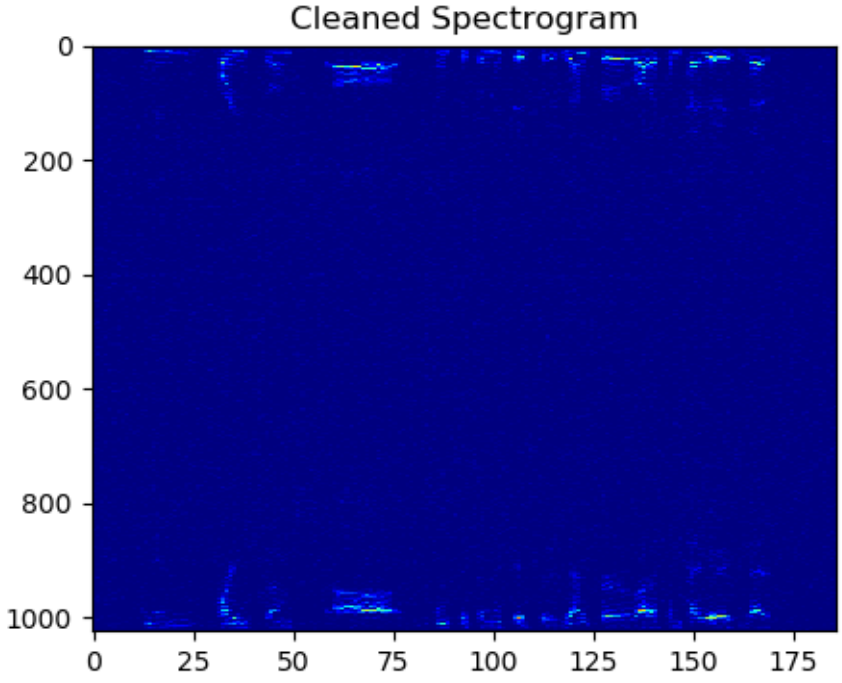
(c) Now we use this DFT matrix and Hann windowed data matrix to get a spectrogram:



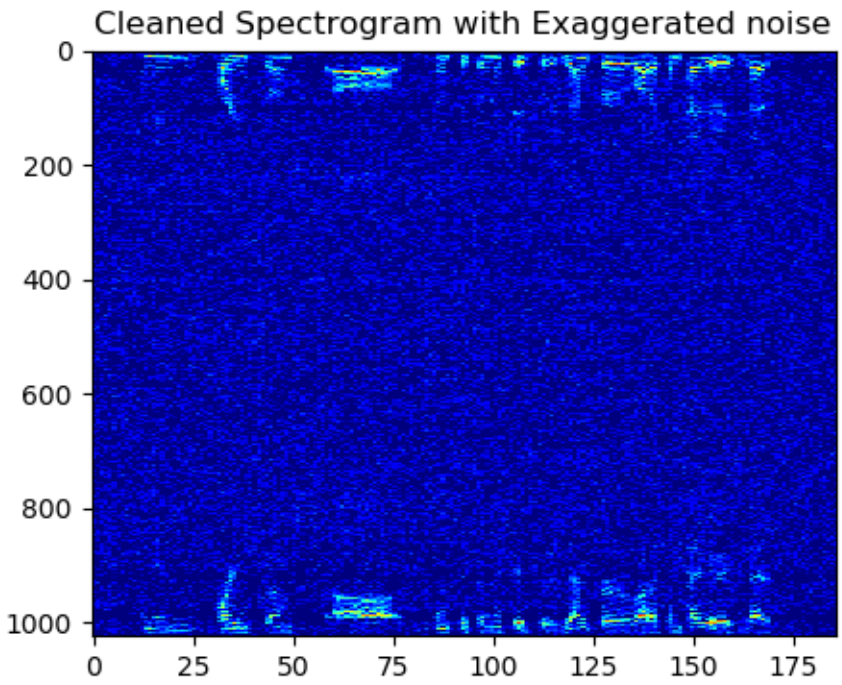
(d) We can see the exaggerated noise - version of this spectrogram:



(e) We can see the last 20 or so columns of this exaggerated noise spectrogram is just noise. We use this information to our advantage and subtract these as the noise from the spectrum. Doing so, we get this cleaned spectrogram:

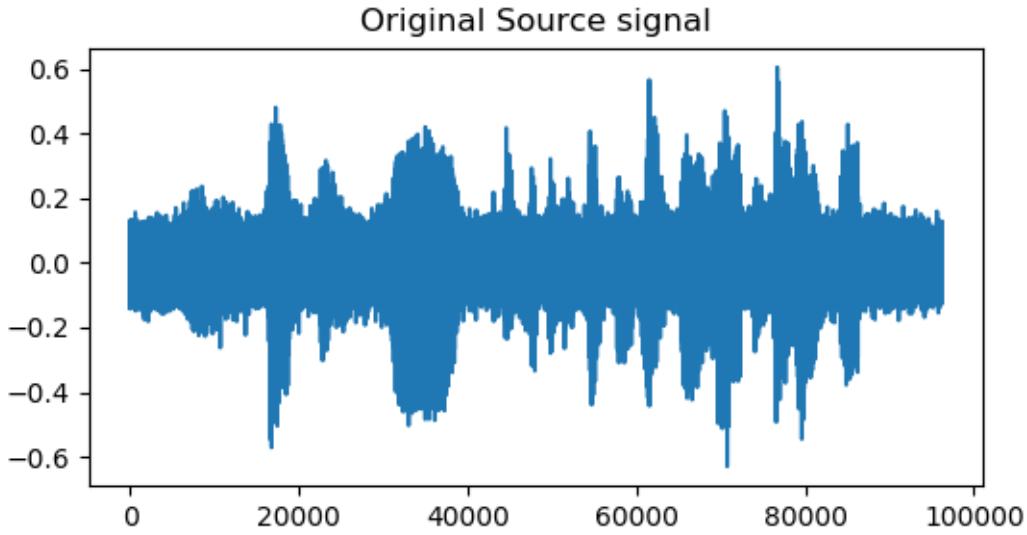


(f) The exaggerated noise - version of this cleaned spectrogram:

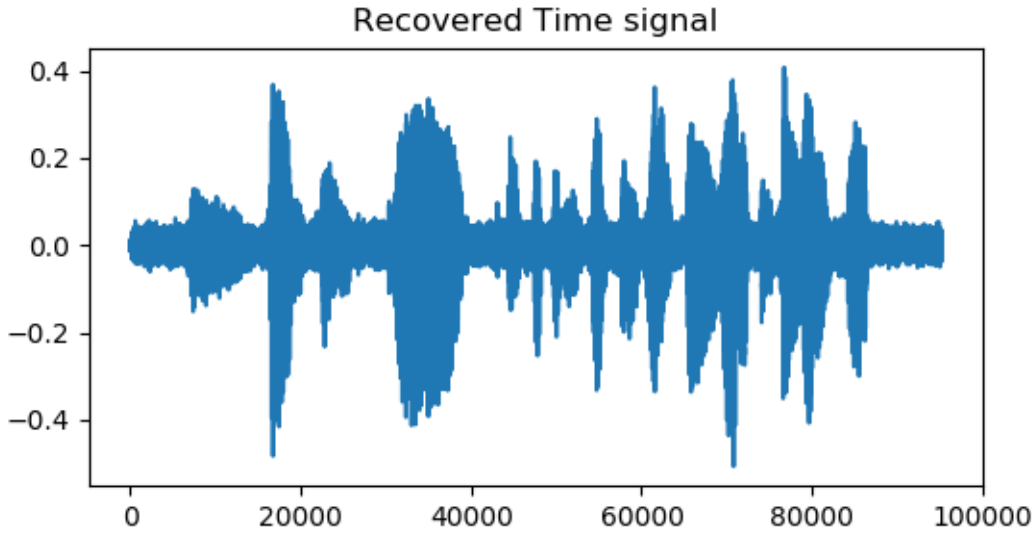


Comparing the exaggerated noise spectrograms in (d) and (f), we can say that the noise has been reduced in (f)

(g) Let's take a look at the original time signal source :



(h) Let's take a look at the recovered time signal after reducing the noise:

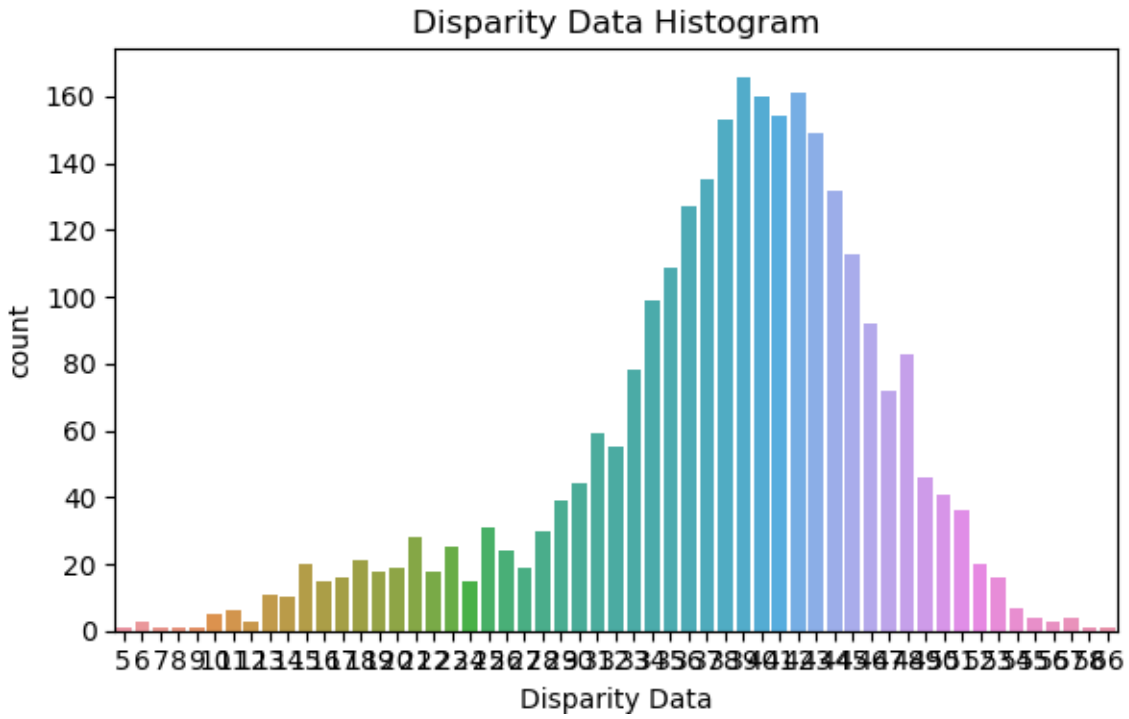


Looking at the recovered time signal and original time signal, we can say that the noise has been reduced to some extent in the cleaned signal as it looks more spiky.

Problem 2: Parallax

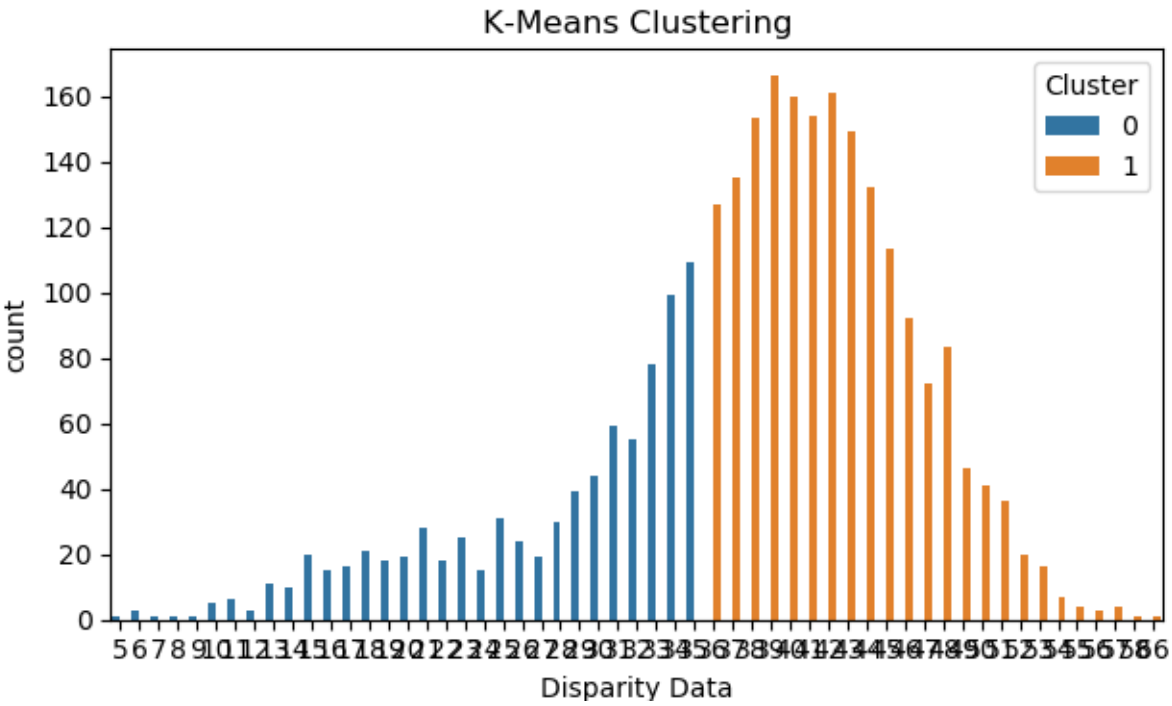
Here we have to use the kmeans algorithm to find out the clusters for disparity matrix. Disparity matrix is the matrix of oscillation values of all the stars in both the galaxies

Here is the histogram of the disparity matrix:



The cluster means found from the k-means clustering are **27.74271845 and 42.31183369**.

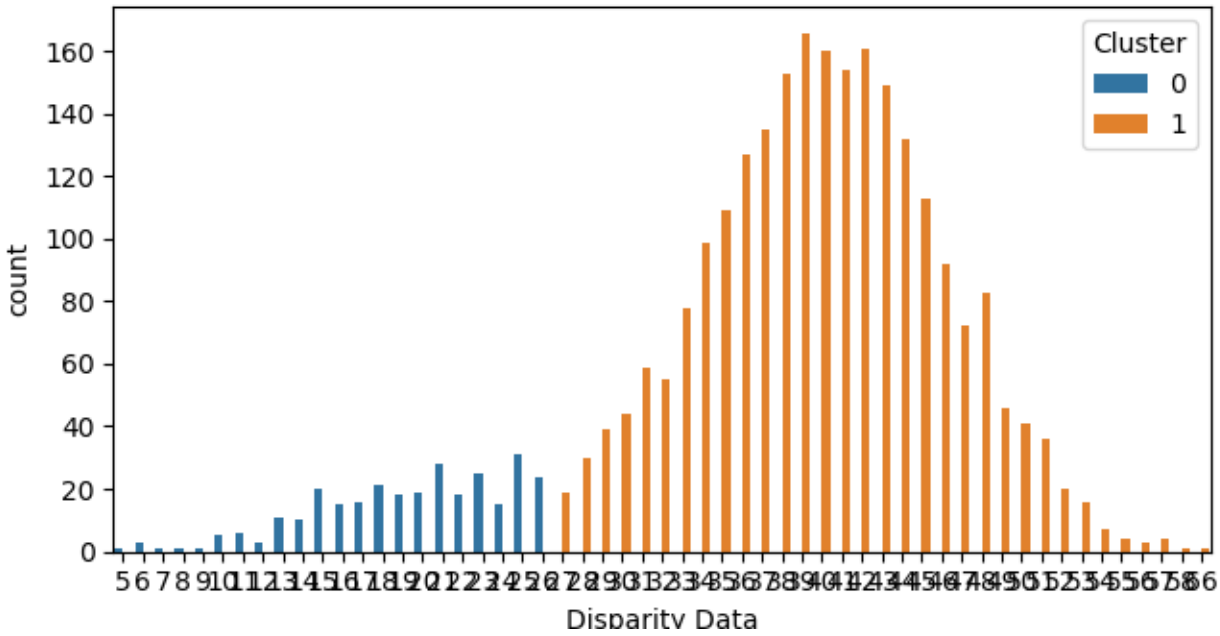
Let's look at the k-means clustering results in plot:



The cluster mean **42.31183369** corresponds to the stars in our galaxy as the oscillations observed for the stars in our galaxy will be more compared to the ones from the other galaxy.

Problem 3: GMM for Parallax

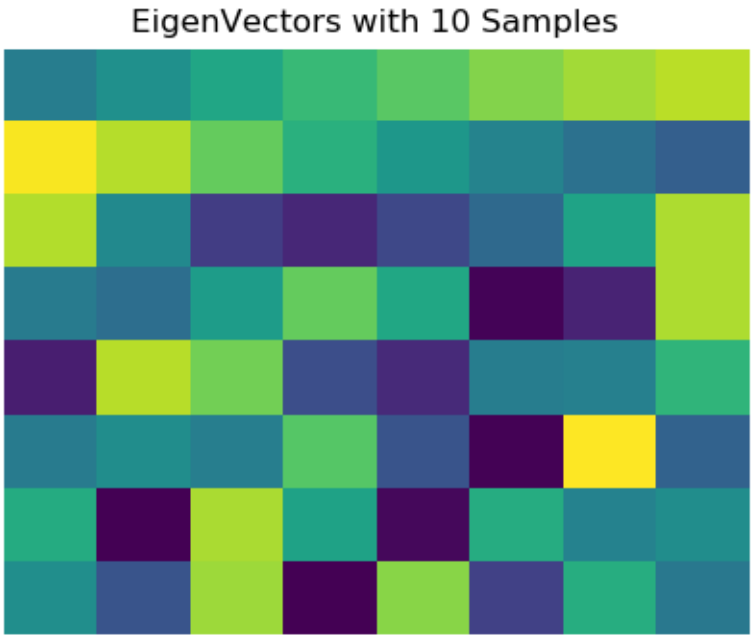
Implementing the **Expectation Maximization(EM)** algorithm for the previous problem.
The means found by EM algorithm are **20.843233835978737, 40.15282866710127** and their respective **standard deviations** are **5.870210400534217 and 5.843101513473928**



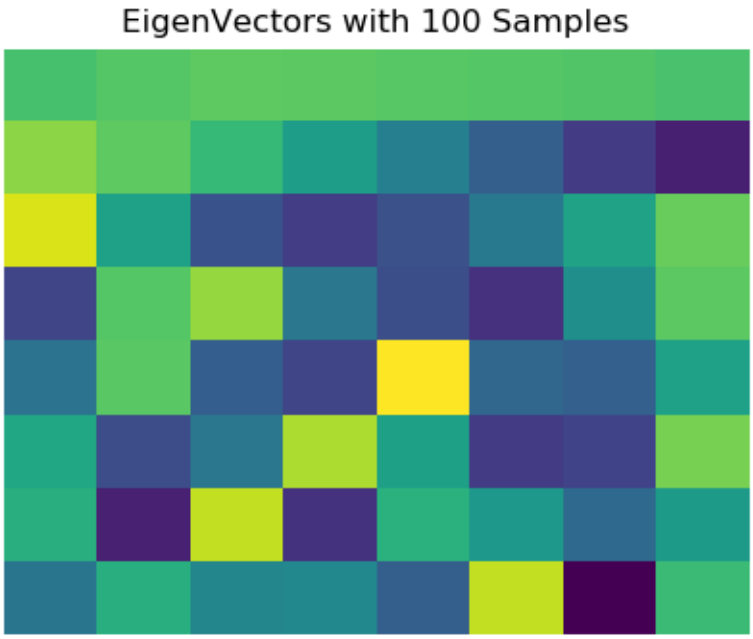
If we look at the result of the EM clustering, we can see that the EM clustering has nicely separated clusters in two different Gaussians.
However in k-means clustering, some of the stars with greater oscillation values have been assigned to the cluster with the lower mean(27.74) just because the mean was closer to those values.
This is because the k-means clustering takes a hard decision whereas GMM takes into account the probabilities of all the data points. Hence it takes a soft decision while assigning the cluster to the data point.
Looking at the original histogram of the disparity data, we can clearly see the two Gaussians.
Hence EM algorithm is preferred here as it does a better job in allocating clusters compared to K-means.

Problem 4: DCT and PCA

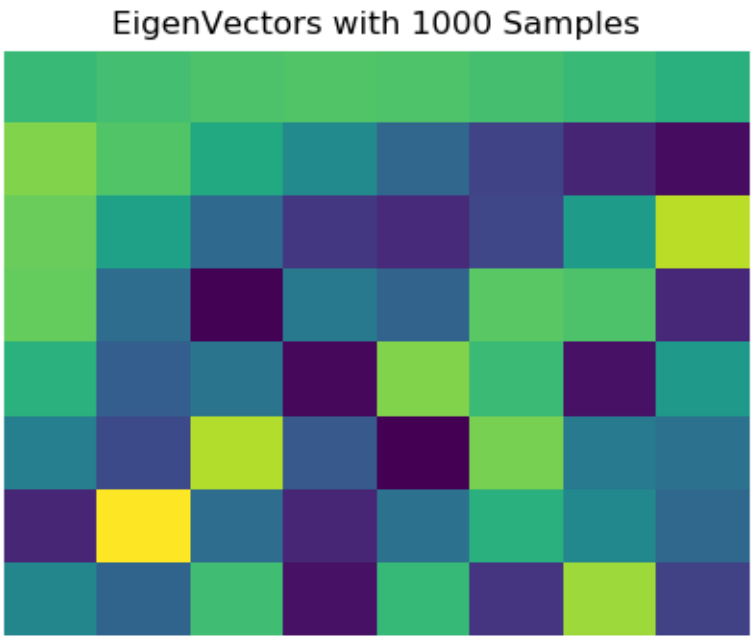
(a) PCA with 10 data samples



(b) PCA with 100 data samples



(c) PCA with 1000 data samples



From the above figures we can say that as the number of data samples increase, PCA gets better and looks closer to the DCT matrix. This is because as the number of data samples are increased, the covariance matrix takes into account the covariance of the larger data. In other words, the covariance matrix comes closer to the true covariance matrix as the data samples are increased and doing an eigendecomposition on this covariance matrix will give the better eigenvectors covering the maximum variance of the data under consideration.

(d) Pros and cons of PCA

Pros of PCA:
PCA is done on the covariance matrix. Hence it is dependent on the underlying data.
PCA gives the eigenvectors(principle components) which represents the most variance of the underlying data. Given the sample data, taken into consideration for PCA, is large enough, PCA is less lossy
Cons of PCA:
We need to calculate the covariance matrix for PCA, and it can be computationally heavy if the data is very high dimensional
For the reconstruction of the data, we need both the eigenvectors as well as the data represented in this eigenvectors-space.
We need large data sample size for the accurate covariance matrix and if the sample size is smaller, the principle components might not cover most variance of the actual data.(As we have seen in the sample data size of 10)

Pros of DCT:
DCT is not dependent on data unlike PCA and is precalculated. Hence it is computationally lighter compared to PCA.

Cons of PCA:
As DCT does not depend on the underlying data and is pre calculated, it might not capture most of the variance of the given data. Hence the data loss might be more in DCT.
The assumption that given data follows a cosine distribution, is harsh and might not be the case always.

References