

Robert Roche

CS383 HW3

Problem 1: Theory (more details in the python notebook)

$$\begin{pmatrix} -2 & 1 \\ -5 & -4 \\ -3 & 1 \\ 0 & 3 \\ -8 & 11 \\ -2 & 5 \\ 1 & 0 \\ 5 & -1 \\ -1 & -3 \\ 6 & 1 \end{pmatrix}$$

- a. Mean x = sum column1 / 10 = -0.9
Mean y = sum column2 / 10 = 1.4

$$\text{Std_matrix} \begin{pmatrix} -2 + 0.9 & 1 - 1.4 \\ -5 + 0.9 & -4 - 1.4 \\ -3 + 0.9 & 1 - 1.4 \\ 0 + 0.9 & 3 - 1.4 \\ -8 + 0.9 & 11 - 1.4 \\ -2 + 0.9 & 5 - 1.4 \\ 1 + 0.9 & 0 - 1.4 \\ 5 + 0.9 & -1 - 1.4 \\ -1 + 0.9 & -3 - 1.4 \\ 6 + 0.9 & 1 - 1.4 \end{pmatrix} = \begin{pmatrix} -1.1 & -0.4 \\ -4.1 & -5.4 \\ -2.1 & -0.4 \\ 0.9 & 1.6 \\ -7.1 & 9.6 \\ -1.1 & 3.6 \\ 1.9 & -1.4 \\ 5.9 & -2.4 \\ -0.1 & -4.4 \\ 6.9 & -0.4 \end{pmatrix}$$

Using (new_mat.T @ new_mat)/10 we get: Covariance Matrix $\begin{bmatrix} 16.09 & -6.64 \\ -6.64 & 16.44 \end{bmatrix}$

We can then use the formula:

$$\lambda^2 - \lambda(a_{11} + a_{22}) + (a_{11}a_{22} - a_{12}a_{21}) = 0$$

$$\lambda^2 - \lambda(16.09 + 16.44) + ((16.09*16.44) - (-6.64*-6.64)) = 0$$

$$\lambda^2 - 32.53\lambda + 220.43 = 0$$

Using the quadratic formula we get the following roots:

$$\text{Eigen values} = 9.623, 22.9073$$

Our Eigen vectors are then:

PC1=[-0.71636104,0.69772979]

PC2=[0.69772979, -0.71636104]

b. Using numpy

X @ PrincipleComponents.T =

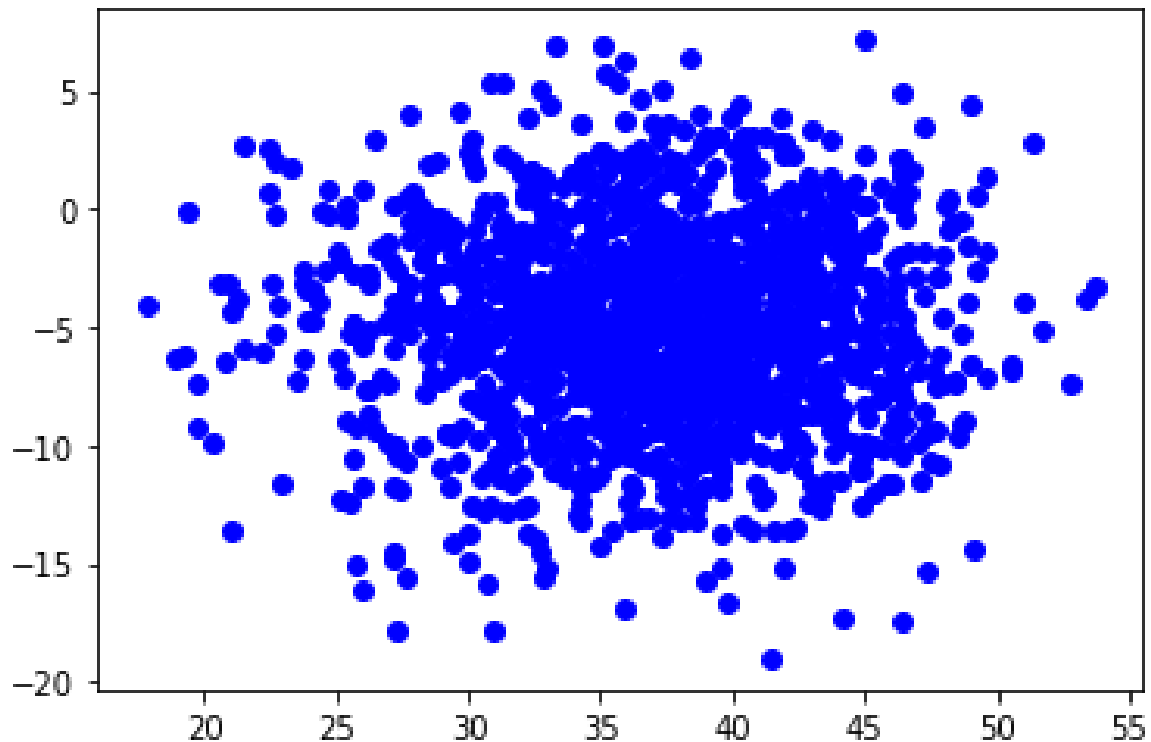
$$\begin{bmatrix} 0.50890522 & 1.05404719 \\ -0.83066062 & 6.72904175 \\ 1.22526626 & 1.75177698 \\ 0.47164273 & -1.77413447 \\ 11.78436938 & -1.92318444 \\ 3.29982439 & -1.81139696 \\ -2.33790768 & -0.32278115 \\ -5.90108163 & -2.39733929 \\ -2.99837498 & 3.22176155 \\ -5.22198308 & -4.52779115 \end{bmatrix}$$

Problem 2:

Upon viewing the code, I believe my KNN and KNN after PCA are correct, my Whitening may have had an issue as I did the necessary computations but found the same results. As described in the slack I graphed X_{train} with 2 dimensions.

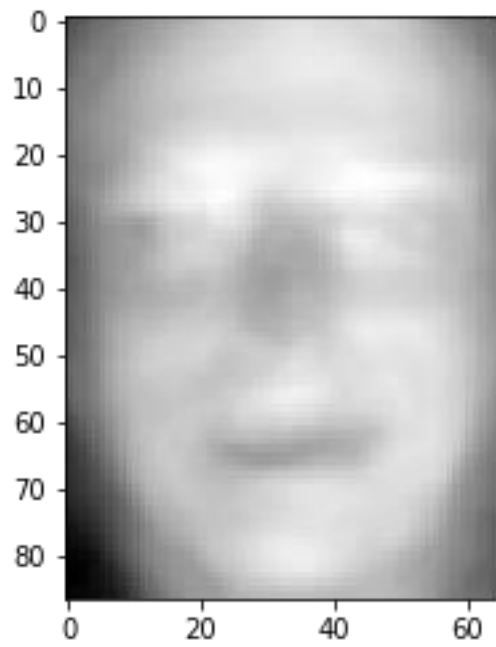
Note The code takes some time to run. Please give it a while to execute, there are no infinite loops.

```
Test set score of 1-nn :0.23
Accuracy of my KNN implementation: 0.23
Accuracy of my KNN implementation after PCA: 0.23255813953488372
Accuracy of my KNN implementation after Whitening+PCA: 0.23255813953488372
```



Problem 3:

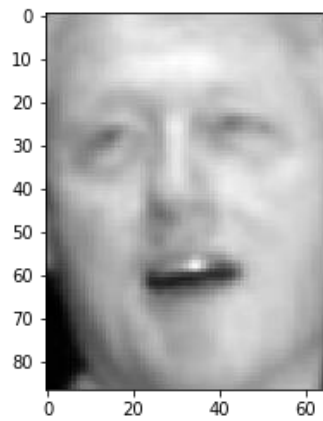
- i. Primary Principle Component Image



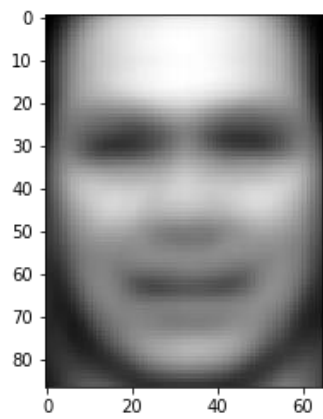
- ii. 5733 Principle Components needed for 95% of the info

iii.

A. Original Image



B. Single Principle construction



C. 95% Reconstruction (k-principle components)

