ISYE 6740 – Midterm 1
Nicholas Chao – nchao3@gatech.edu

1. K-Means

Euclidean Distance between two vectors, x_1 and x_2: $\sqrt{\sum(x_1 - x_2)^2}$
Manhattan Distance between two vectors, x_1 and x_2: $\sum|x_1 - x_2|$

Geometric Mean of set of N vectors: $\frac{\sum_n^N x_n}{N}$

Initial Centroid A: $(-3, -1)$
Initial Centroid B: $(2, 1)$

Using Euclidean Distance:

(a) Datapoint 1 -> Cluster B
Datapoint 2 -> Cluster A
Datapoint 3 -> Cluster B
Datapoint 4 -> Cluster B
Datapoint 5 -> Cluster A

Datapoint 1: $(2, 2)$

Distance from A: $\sqrt{\sum\left((2,2) - (-3,-1)\right)^2} = \sqrt{\sum(5,3)^2} = \sqrt{\sum(25,9)} = \sqrt{34} = 5.83$

Distance from B: $\sqrt{\sum\left((2,2) - (2,1)\right)^2} = \sqrt{\sum(0,1)^2} = \sqrt{\sum(0,1)} = \sqrt{1} = \mathbf{1}$

Datapoint 2: $(-1, 1)$

Distance from A: $\sqrt{\sum\left((-1,1) - (-3,-1)\right)^2} = \sqrt{\sum(2,1)^2} = \sqrt{\sum(4,1)} = \sqrt{5} = \mathbf{2.24}$

Distance from B: $\sqrt{\sum\left((-1,1) - (2,1)\right)^2} = \sqrt{\sum(-3,0)^2} = \sqrt{\sum(9,0)} = \sqrt{9} = 3$

Datapoint 3: $(3, 1)$

Distance from A: $\sqrt{\sum\left((3,1) - (-3,-1)\right)^2} = \sqrt{\sum(6,2)^2} = \sqrt{\sum(36,4)} = \sqrt{40} = 6.32$

Distance from B: $\sqrt{\sum\left((3,1) - (2,1)\right)^2} = \sqrt{\sum(1,0)^2} = \sqrt{\sum(1,0)} = \sqrt{1} = \mathbf{1}$

Datapoint 4: $(0, -1)$

Distance from A: $\sqrt{\sum\left((0,-1) - (-3,-1)\right)^2} = \sqrt{\sum(3,0)^2} = \sqrt{\sum(9,0)} = \sqrt{9} = 3$

Distance from B: $\sqrt{\sum\left((0,-1)-(2,1)\right)^2} = \sqrt{\sum(-2,-2)^2} = \sqrt{\sum(4,4)} = \sqrt{8} = \mathbf{2.83}$

Datapoint 5: $(-2,-2)$

Distance from A: $\sqrt{\sum\left((-2,-2)-(-3,-1)\right)^2} = \sqrt{\sum(1,-1)^2} = \sqrt{\sum(1,1)} = \sqrt{2} = \mathbf{1.41}$

Distance from B: $\sqrt{\sum\left((-2,-2)-(2,1)\right)^2} = \sqrt{\sum(-4,-3)^2} = \sqrt{\sum(16,9)} = \sqrt{25} = 5$

(b) Cluster A's centroid becomes (-1.5, -0.5)
Cluster B's centroid becomes (1.667, 0.667)

Cluster A: $\{(-1,1); (-2,-2)\}$

Geometric Mean: $\frac{\sum_n^N x_n}{N} = \frac{(-1,1)+(-2,-2)}{2} = \frac{(-3,-1)}{2} = (-1.5,-0.5)$

Cluster B: $\{(2,2); (3,1); (0,-1)\}$

Geometric Mean: $\frac{\sum_n^N x_n}{N} = \frac{(2,2)+(3,1)+(0,-1)}{3} = \frac{(5,2)}{3} = (1.667, 0.667)$

(c) The algorithm **will not** terminate after one step. Another iteration will result in the follow clusters:

Datapoint 1 -> Cluster B
Datapoint 2 -> Cluster A
Datapoint 3 -> Cluster B
Datapoint 4 -> Cluster A
Datapoint 5 -> Cluster A

Datapoint 1: $(2,2)$

Distance from A: 4.30

Distance from B: $\mathbf{1.37}$

Datapoint 2: $(-1,1)$

Distance from A: $\mathbf{1.58}$

Distance from B: 2.69

Datapoint 3: $(3,1)$

Distance from A: 4.74

Distance from B: $\mathbf{1.37}$

Datapoint 4: $(0, -1)$

Distance from A: $\mathbf{1.58}$

Distance from B: 2.36

Datapoint 5: $(-2, -2)$

Distance from A: $\mathbf{1.58}$

Distance from B: 4.53

Using Manhattan Distance:

(a) Datapoint 1 -> Cluster B
Datapoint 2 -> Cluster B
Datapoint 3 -> Cluster B
Datapoint 4 -> Cluster A
Datapoint 5 -> Cluster A

Datapoint 1: $(2, 2)$

Distance from A: $\sum|(2,2) - (-3,-1)| = \sum|(5,3)| = \sum(5,3) = 8$

Distance from B: $\sum|(2,2) - (2,1)| = \sum|(0,1)| = \sum(0,1) = \mathbf{1}$

Datapoint 2: $(-1, 1)$

Distance from A: $\sum|(-1,1) - (-3,-1)| = \sum|(2,2)| = \sum(2,2) = 4$

Distance from B: $\sum|(-1,1) - (2,1)| = \sum|(-3,0)| = \sum(3,0) = \mathbf{3}$

Datapoint 3: $(3, 1)$

Distance from A: $\sum|(3,1) - (-3,-1)| = \sum|(6,2)| = \sum(6,2) = 8$

Distance from B: $\sum|(3,1) - (2,1)| = \sum|(1,0)| = \sum(1,0) = \mathbf{1}$

Datapoint 4: $(0, -1)$

Distance from A: $\sum|(0,-1) - (-3,-1)| = \sum|(3,0)| = \sum(3,0) = \mathbf{3}$

Distance from B: $\sum|(0,-1) - (2,1)| = \sum|(-2,-2)| = \sum(2,2) = 4$

Datapoint 5: (-2, -2)

Distance from A: $\sum|(-2,-2) - (-3,-1)| = \sum|(1,-2)| = \sum(1,1) = \mathbf{2}$

Distance from B: $\sum|(-2,-2) - (2,1)| = \sum|(-4,-3)| = \sum(4,3) = 7$

(b) Cluster A’s centroid becomes (-1, -1.5)

Cluster B's centroid becomes (1.33, 1.33)

Cluster A: $\{(0,-1); (-2,-2)\}$

Geometric Mean: $\frac{\sum_n^N x_n}{N} = \frac{(0,-1)+(-2,-2)}{2} = \frac{(-2,-3)}{2} = (-1,-1.5)$

Cluster B: $\{(2,2); (-1,1); (3,1)\}$

Geometric Mean: $\frac{\sum_n^N x_n}{N} = \frac{(2,2)+(-1,1)+(3,1)}{3} = \frac{(4,4)}{3} = (1.33, 1.33)$

(c) The algorithm **will not** terminate after one step. Another iteration will result in the follow clusters:

Datapoint 1 -> Cluster B
Datapoint 2 -> Cluster A
Datapoint 3 -> Cluster B
Datapoint 4 -> Cluster A
Datapoint 5 -> Cluster A

Datapoint 1: $(2,2)$

Distance from A: 6.5

Distance from B: $\mathbf{1.33}$

Datapoint 2: $(-1,1)$

Distance from A: $\mathbf{2.5}$

Distance from B: 2.67

Datapoint 3: $(3,1)$

Distance from A: 6.5

Distance from B: $\mathbf{2}$

Datapoint 4: $(0,-1)$

Distance from A: $\mathbf{1.5}$

Distance from B: 3.67

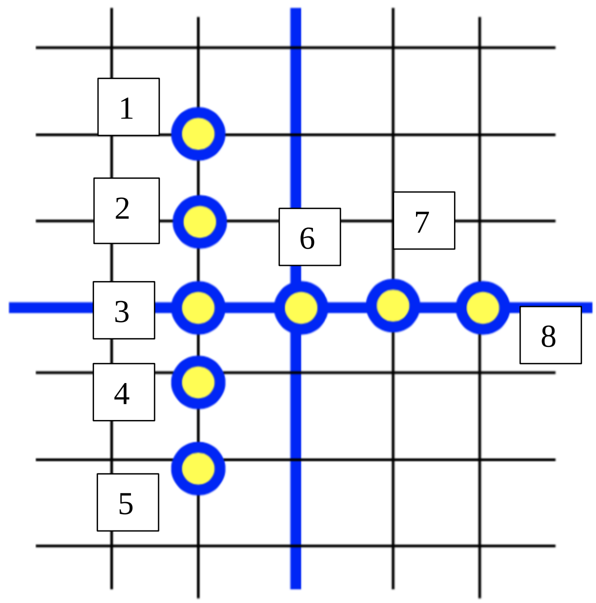
Datapoint 5: $(-2,-2)$

Distance from A: $\mathbf{1.5}$

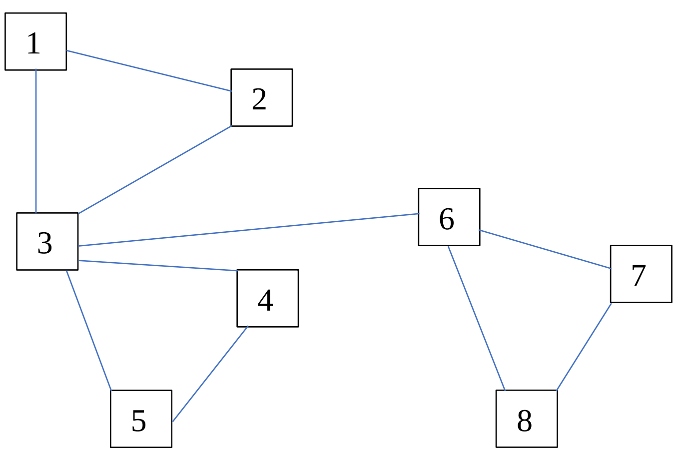
Distance from B: 6.67

2. Spectral Clustering (R code for eigen-decomposition calculations included separately in Q2.R)

Give the following labels for each point in the data:



The nearest-neighbor embedding will result in the following graph:



Edge weights are then calculated by:

$$W_{ij} = exp(-\|x_i - x_j\|)$$

Using the above weight formula, the following Adjacency matrix for the neighborhood graph is built:

$$\begin{bmatrix} 0 & 0.3678794 & 0.1353353 & 0 & 0 & 0 & 0 & 0 \\ 0.3678794 & 0 & 0.3678794 & 0 & 0 & 0 & 0 & 0 \\ 0.1353353 & 0.3678794 & 0 & 0.3678794 & 0.1353353 & 0.3678794 & 0 & 0 \\ 0 & 0 & 0.3678794 & 0 & 0.3678794 & 0 & 0 & 0 \\ 0 & 0 & 0.1353353 & 0.3678794 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.3678794 & 0 & 0 & 0 & 0.3678794 & 0.1353353 \\ 0 & 0 & 0 & 0 & 0 & 0.3678794 & 0 & 0.3678794 \\ 0 & 0 & 0 & 0 & 0 & 0.1353353 & 0.3678794 & 0 \end{bmatrix}$$

Where each row and column correspond to each of the 8 datapoints in order.

For example, $A_{21} = 0.3678794$ because $X_2 = \{-1, 1\}$ and $X_1 = \{-1, 2\}$. Therefore,

$$W_{21} = exp(-\|x_2 - x_1\|) = exp(-\|\{-1, 1\} - \{-1, 2\}\|) = exp(-1) = 0.3678794.$$

Similarly, $A_{31} = 0.1353353$ because $X_3 = \{-1, 0\}$ and $X_1 = \{-1, 2\}$. Therefore,

$$W_{31} = exp(-\|x_3 - x_1\|) = exp(-\|\{-1, 0\} - \{-1, 2\}\|) = exp(-2) = 0.1353353.$$

These calculations were done for each datapoint that shared an edge in the neighborhood graph to build the adjacency matrix.

The Degree matrix is then built, using $D_{ii} = \sum_e^E W_{ie}$ where E represents all the edges of node i. This results in the following matrix:

$$\begin{bmatrix} 0.5032147 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.7357589 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1.374309 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.7357589 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.5032147 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.8710942 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0.7357589 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.5032147 \end{bmatrix}$$

For example, $D_{11} = 0.5032147$ because Node 1 has edges with Nodes 2 and 3, with weights 0.3678794 and 0.1353353, respectively. These weights sum up to 0.5032147 to give that node's weighted degree.

The Laplacian is then calculated as $L = D - A$. This gives the following Laplacian matrix:

$$\begin{bmatrix} 0.5032147 & -0.3678794 & -0.1353353 & 0 & 0 & 0 & 0 & 0 \\ -0.3678794 & 0.7357589 & -0.3678794 & 0 & 0 & 0 & 0 & 0 \\ -0.1353353 & -0.3678794 & 1.374309 & -0.3678794 & -0.1353353 & -0.3678794 & 0 & 0 \\ 0 & 0 & -0.3678794 & 0.7357589 & -0.3678794 & 0 & 0 & 0 \\ 0 & 0 & -0.1353353 & -0.3678794 & 0.5032147 & 0 & 0 & 0 \\ 0 & 0 & -0.3678794 & 0 & 0 & 0.8710942 & -0.3678794 & -0.1353353 \\ 0 & 0 & 0 & 0 & 0 & -0.3678794 & 0.7357589 & -0.3678794 \\ 0 & 0 & 0 & 0 & 0 & -0.1353353 & -0.3678794 & 0.5032147 \end{bmatrix}$$

The eigen-decomposition is then found for this Laplacian. This results in the following Eigenvalues and corresponding Eigenvectors:

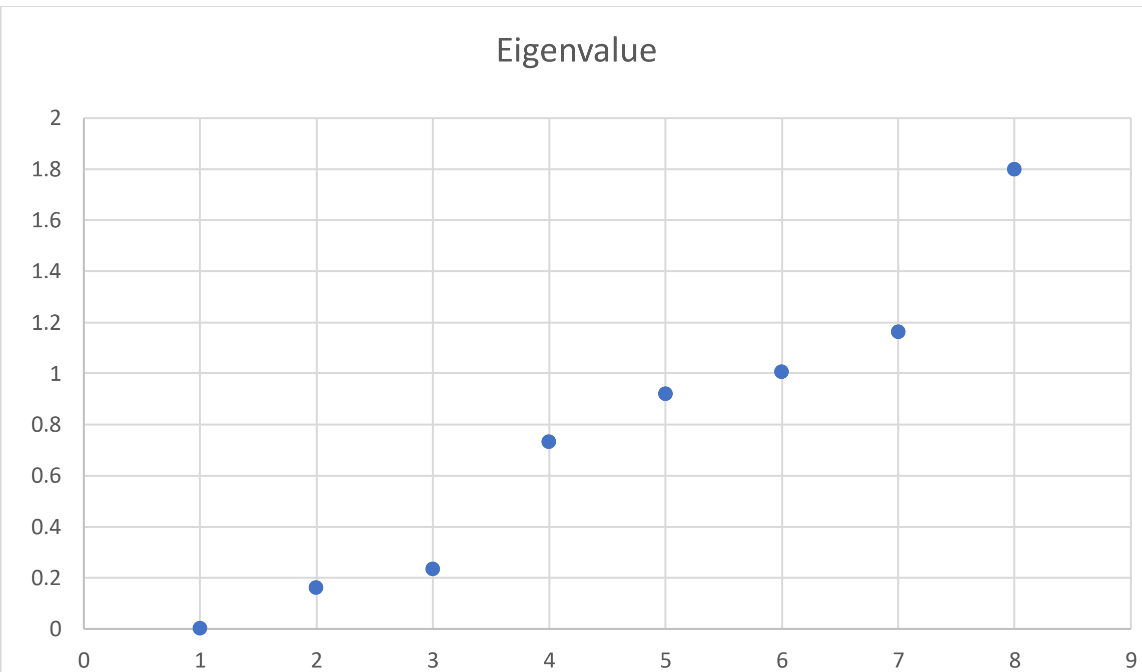
Eigenvalues:

$$\{1.798349, 1.160359, 1.005303, 0.9183616, 0.7302276, 0.2336702, 0.160535, 0\}$$

Eigenvectors (represented at $p \times k$ matrix where p is number of dimensions of eigenvectors and k is the number of eigenvectors):

$$\begin{bmatrix} -0.005624343 & 0.09854295 & 0.4179216 & 0.4231428 & -0.29717638 & 0.5703872 & 0.3128803 & 0.3535534 \\ -0.283299265 & -0.24291919 & -0.5703872 & -0.3937439 & 0.07365058 & 0.4179216 & 0.2743595 & 0.3535534 \\ 0.823911178 & 0.18183020 & 3.941292e-15 & -0.2277019 & 0.29828377 & 2.498002e-16 & 0.1492877 & 0.3535534 \\ -0.283299265 & -0.24291919 & 0.5703872 & -0.3937439 & 0.07365058 & -0.4179216 & 0.2743595 & 0.3535534 \\ -0.005624343 & 0.09854295 & -0.4179216 & 0.4231428 & -0.29717638 & -0.5703872 & 0.3128803 & 0.3535534 \\ -0.378952448 & 0.51908263 & -2.470246e-15 & 0.1939445 & 0.59358261 & -1.387779e-15 & -0.2683160 & 0.3535534 \\ 0.130301606 & -0.69347736 & -7.660539e-15 & 0.3353704 & 0.14656414 & -1.249001e-15 & -0.4931088 & 0.3535534 \\ 0.002586879 & 0.28131702 & 7.327472e-15 & -0.3604109 & -0.59137891 & -1.935951e-15 & -0.5623425 & 0.3535534 \end{bmatrix}$$

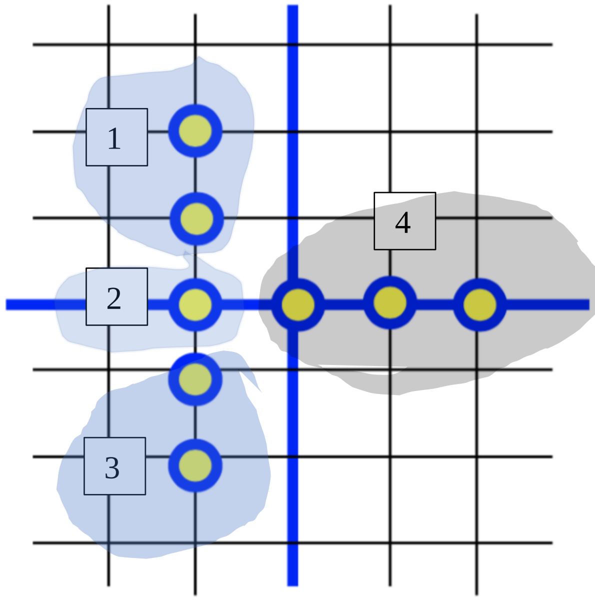
I then decided to use the smallest 2 non-zero eigenvalues for the spectral cluster algorithm, as a noticeable gap is observed between the 2nd smallest non-zero eigenvalue and the third;



This leaves me with the following matrix (taking the corresponding eigenvectors):

$$\begin{bmatrix} 0.5703872 & 0.3128803 \\ 0.4179216 & 0.2743595 \\ 2.498002e-16 & 0.1492877 \\ -0.4179216 & 0.2743595 \\ -0.5703872 & 0.3128803 \\ -1.387779e-15 & -0.2683160 \\ -1.249001e-15 & -0.4931088 \\ -1.935951e-15 & -0.5623425 \end{bmatrix}$$

Finally, I ran the K-means algorithm with K = 4 and ended up with the following cluster assignments (treating each row in the above matrix by the starting label assigned, 1 through 8):



3. Principal Component Analysis (See separate R code Q3.R for eigen decomposition calculation on part a)

(a) Find the first principal direction

The first principal direction is given by the eigenvector of the covariance matrix (of the min-max normalized data) that corresponds to the largest eigenvalue. This eigenvector is $\{0.577, -0.577, 0.577\}$.

First, I min-max normalized the data. I then calculated the covariance matrix which is equal to

$$C = \frac{1}{m}\sum_{i=1}^{m}(x_i - \mu)\,(x_i - \mu)^T$$

where μ is the mean vector, $\frac{1}{m}\sum_{i=1}^{m} x_i$.

This results in the following covariance matrix (on the min-max normalized matrix):

$$\begin{bmatrix} 0.1851852 & -0.1851852 & 0.1851852 \\ -0.1851852 & 0.1851852 & -0.1851852 \\ 0.1851852 & -0.1851852 & 0.1851852 \end{bmatrix}$$

The eigen-decomposition of this covariance matrix yields the following eigenvectors and eigenvalues.

Eigenvectors:

$$\begin{bmatrix} 0.5773503 & 0.8164966 & 0 \\ -0.5773503 & 0.4082483 & 0.7071068 \\ 0.5773503 & -0.4082483 & 0.7071068 \end{bmatrix}$$

Eigenvalues:

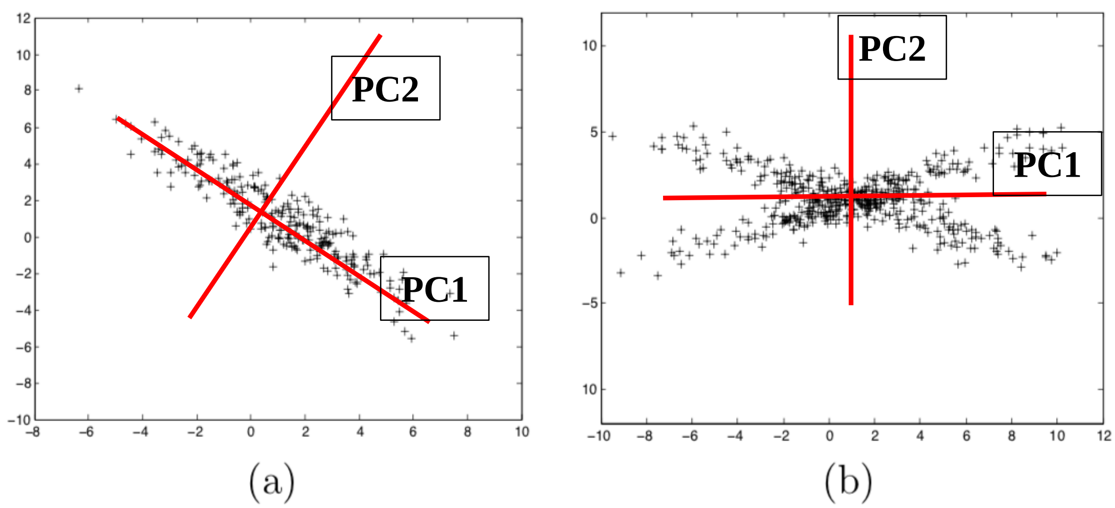
$$\{0.5555556, 2.220446e-16, 0\}$$

The eigenvector that corresponds to the largest eigenvalue is therefore $\{0.577, -0.577, 0.577\}$, giving us the first principal direction in the data.

(b) What is the reconstruction error from the first principal component?

The reconstruction error, in terms of variance found in the data, is given by the eigenvalue. Because the other two eigenvalues are virtually 0, this first principal direction will have a reconstruction error of 0%, or in other words, the first principal component explains 100% of the variance found in the data.

(c) Draw the first and second principal directions in plots.



4. PCA for Face Recognition (Code is in R script, Q4.R)

(1) Generating Eigenfaces

Eigenface 1:

Subject01

Subject14

Eigenface 2:

Subject01

Subject14

Eigenface 3:

Subject01

Subject01

Subject14

Eigenface 4:

Subject01

Subject14

Eigenface 5:

Subject01

Subject14

Eigenface 6:

Subject01

Subject14

(2) Facial Recognition

In order to project the test images against the top eigenfaces, I used cosine-similarity (normalized dot-product) which compares the angle of two vectors. Cosine-similarity is bounded between -1 and 1, and a score of zero means the vectors are orthogonal and thus have no relation. A score of 1 means perfect positive correlation, while a score of -1 means perfect negative correlation.

Score 1 (subject 01 test image vs. subject 01 eigenface 1): -0.9814099
Score 2 (subject 01 test image vs. subject 14 eigenface 1): -0.8720915
Score 3 (subject 14 test image vs. subject 01 eigenface 1): -0.8917421
Score 4 (subject 14 test image vs. subject 14 eigenface 1): -0.9897491

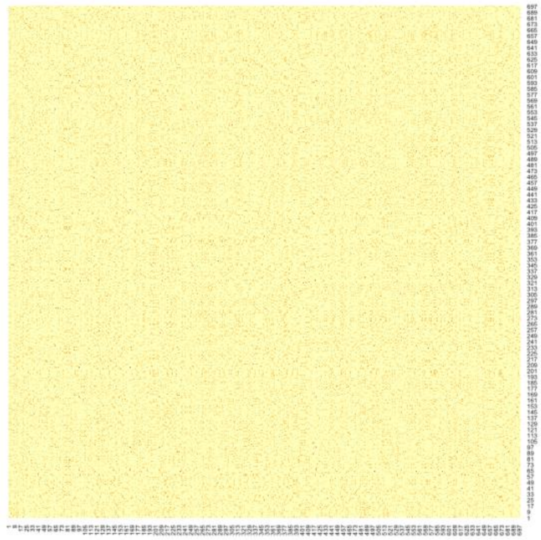
Using these scores we are indeed able to recognize whose face is in each test image. We can see that there is a stronger (negative) relationship between each test image face and their respective top eigenvalue. What is important is not the direction of the relationship, but rather its strength. As stated before, a cosine similarity of 0 indicates orthogonal vectors and therefore no relationship, so we can see that each test subject's test image is closer to being orthogonal to the other subject's top eigenface.

5. Order of Faces using ISOMAP (See R script in Q5.R)

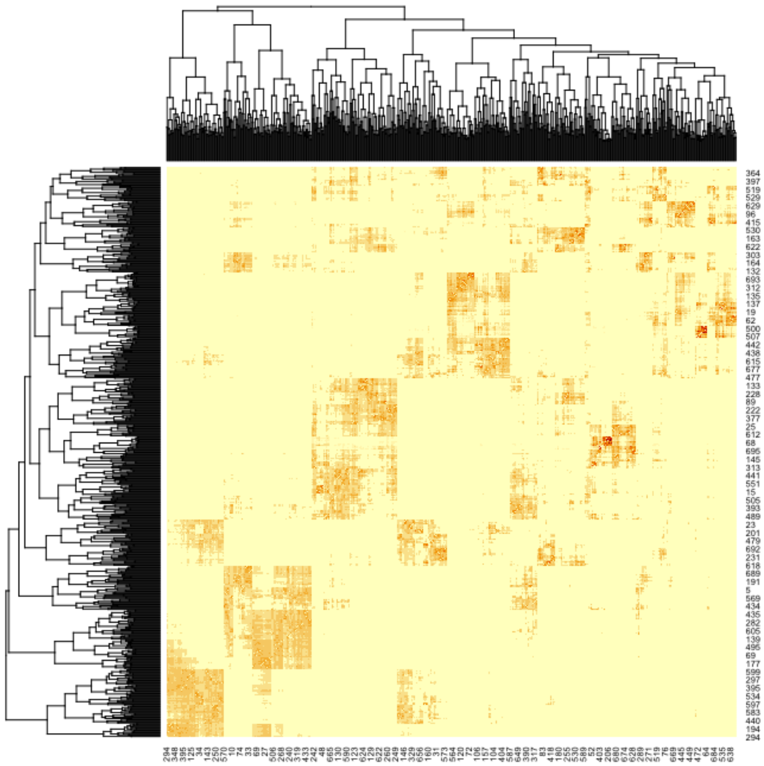
(a) See Q5.R for work.

Visualization of the 100-nearest neighbor graph which creates edges between each image and its 100 nearest neighbors by Euclidean distance:

(Deeper red means higher edge weight, rows and columns are in order of image 1 through 698)

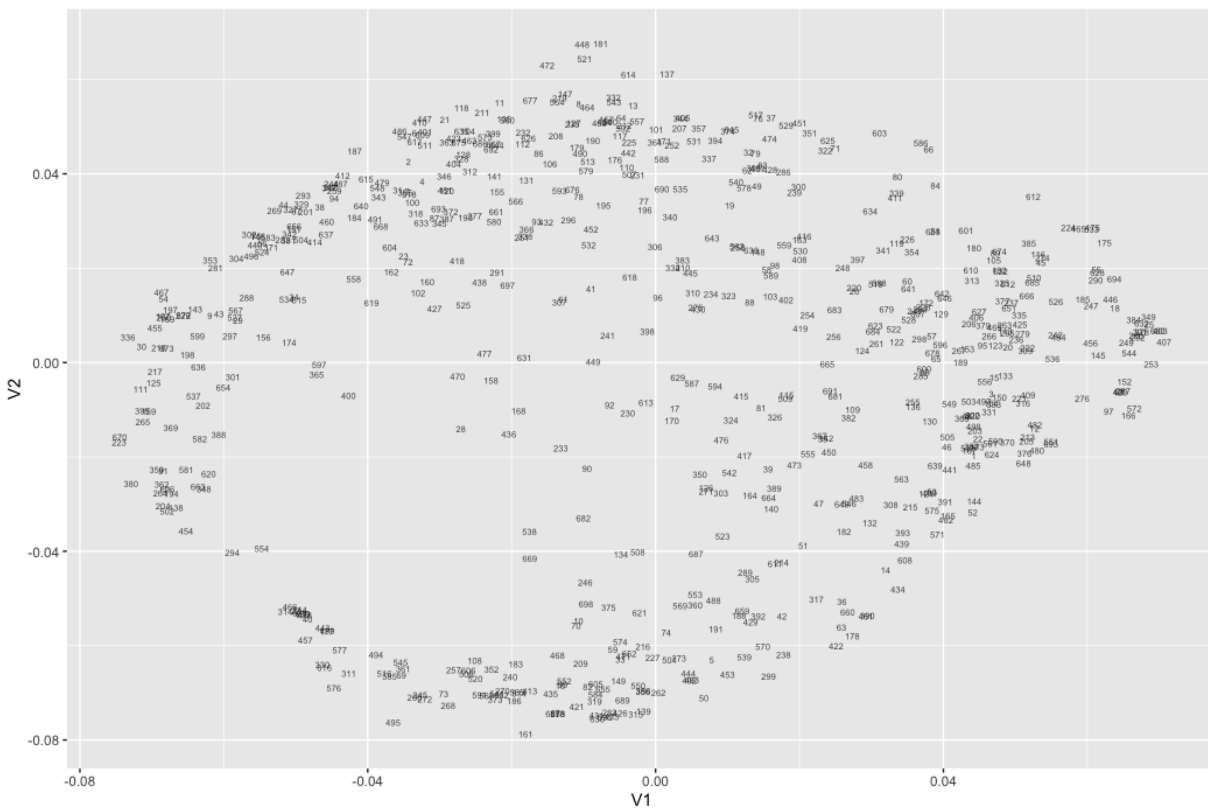


The following plot is the same graph, but with rows and columns reordered based on a hierarchical clustering to help get an idea of the actual shape of the graph:



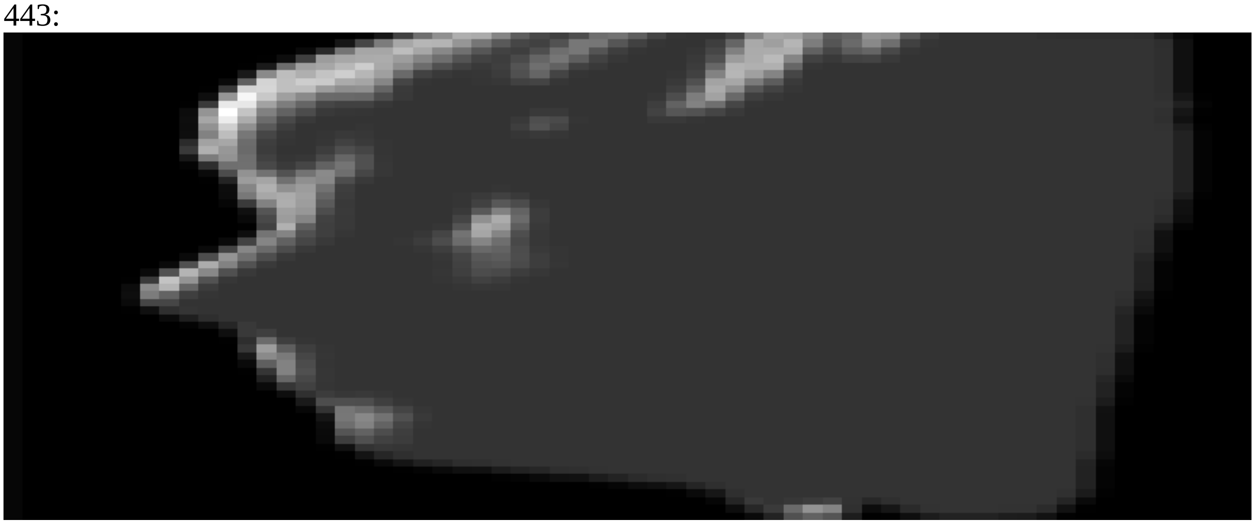
(b) See Q5.R for work.

Plot of 2-dimensional embedding from ISOMAP (numbers represent the image id – taken as the order the images in isomap.mat):

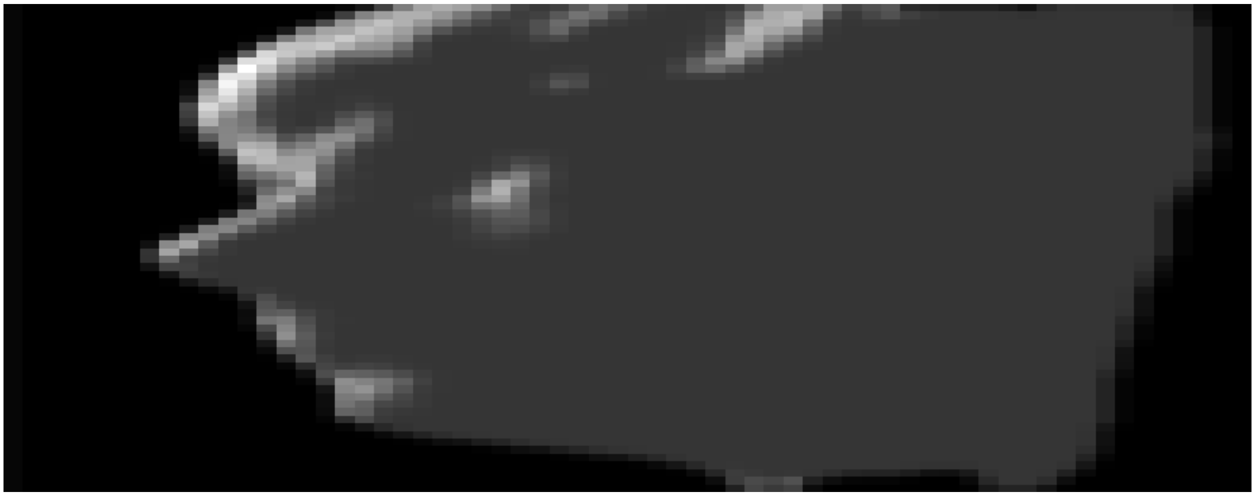


We can see images 443, 177, and 499 are tightly clustered in the bottom left region of the plot. Here are those images:

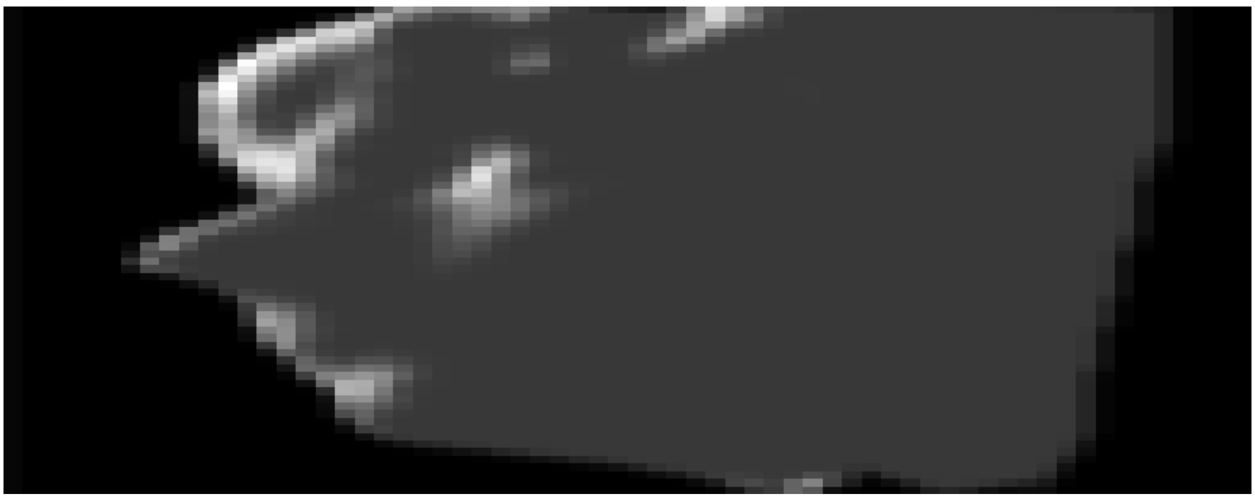
443:



177:



499:



We can see that these three images are extremely similar, all showing faces looking to the left.