ISOMAP and Image Recognition

Hannah Pavlovich

February 11, 2024

Order of faces using ISOMAP

This question aims to reproduce the ISOMAP algorithm results in the original paper for ISOMAP, J.B. Tenenbaum, V. de Silva, and J.C. Langford, Science 290 (2000) 2319-2323.

The file contains 698 images, corresponding to different poses of the same face. Each image is given as a 64×64 luminosity map, hence represented as a vector in $\mathbb{R}^{4096}$. This vector is stored as a row in the file.

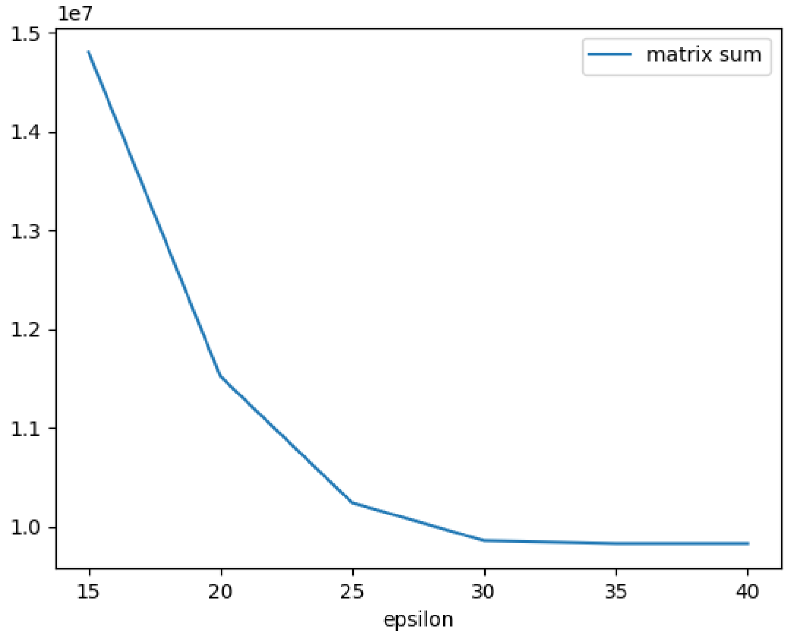
1. Visualize the nearest neighbor graph.

   Reference: https://www.jsoftware.us/vol6/jsw0606-11.pdf
   The equation for the adjacency matrix is:

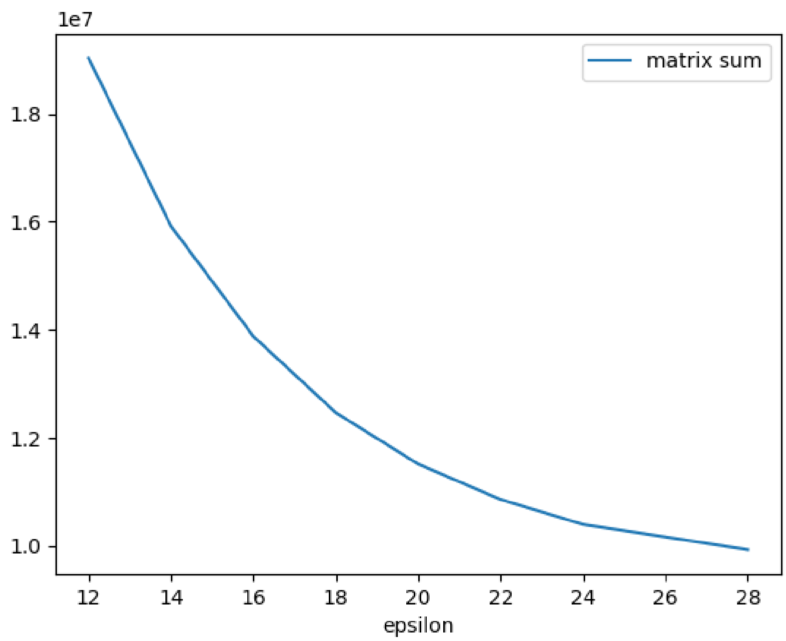
   $$A^{ij} = \begin{Bmatrix} \|x^i - x^j\|, \text{if} \|x^i - x^j\| \leq \epsilon \\ 0, \text{ otherwise} \end{Bmatrix}$$

   In finding the adjacency matrix, it is first needed to find the tuning parameter. To do this, I created multiple adjacency matrices with different epsilons and calculated the sum of the shortest distances. The best epsilon will be the one that has a low sum of all distances. I am looking for the point in the plot before these is a larger jump down.

   I began with a large range to see if there was an obvious choice on the scree plot. The scree plot shows a more gradual transition, with a small elbow around 20. To avoid overfitting, I refocused on this point with a zoomed-in scree plot. This plot also shows a smoother transition rather than a sudden jump. $\epsilon = 16$ is a point before the plot goes to the minimum, so this is the epsilon I will use for this question. Epsilon will be reconsidered in the next part.

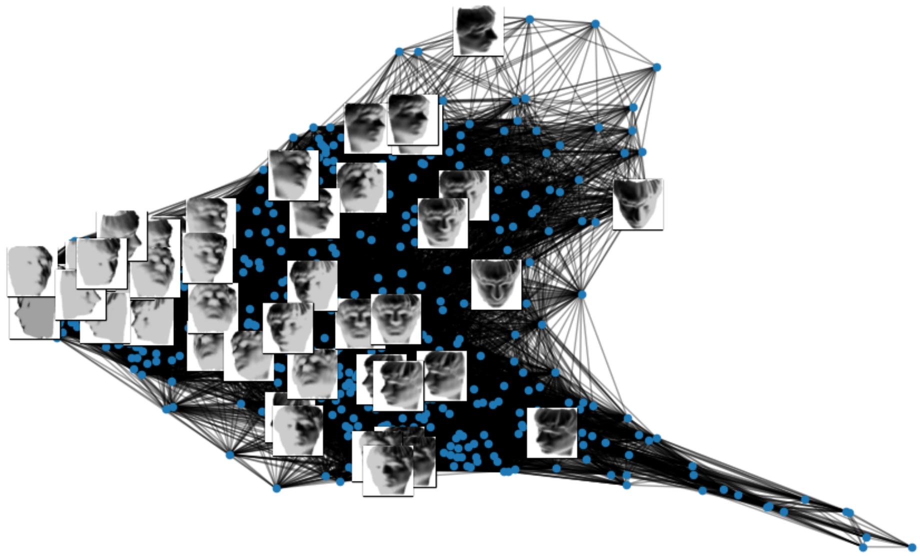


Epsilon vs Distance Matrix Sum Elbow Plot, 15:40



Epsilon vs Distance Matrix Sum Elbow Plot, 12:28

Below is a visualization of this matrix with images of the faces. The lower right of the image has the faces looking down to the left, the upper right has faces looking down and to the right and are darker. The left-most side of the image, however, seems to be mainly grouped around color, as the faces are pointing in different directions but are all light-colored.



Nearest Neighbor Visualization

2. Implement the ISOMAP algorithm yourself to obtain a two-dimensional low-dimensional embedding and plot the embeddings using a scatter plot.

   The steps I took to build the ISOMAP are as follows:

   (a) Build a weighted graph A using the weighted nearest neighbor method

   (b) Compute the pairwise shortest difference between the data points.

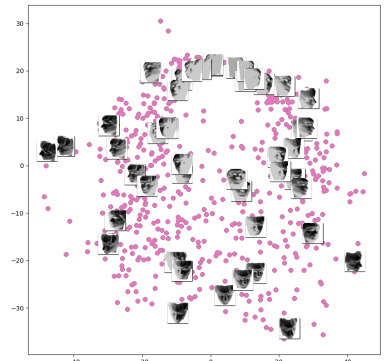
   (c) Use the centering matrix to compute the C-matrix, which captures the pairwise distance for the datapoints

   $$H = 1 - \frac{1}{m}11^T$$

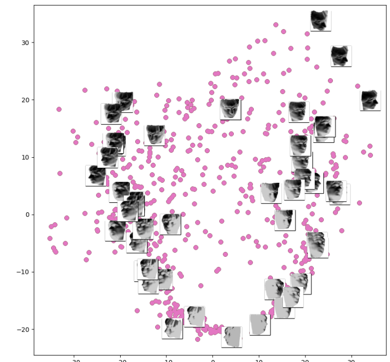
   $$C = -\frac{1}{2}H(D_{ij}^2)H \in R^{mxm}$$

   (d) Compute the eigen-decomposition of the C matrix.

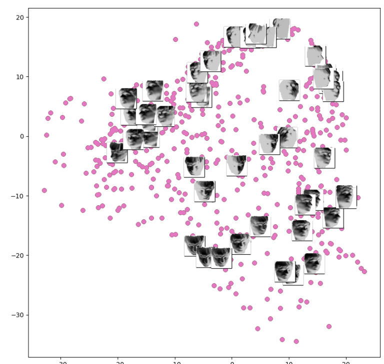
   In this step, I wanted to look at the tuning parameters again,



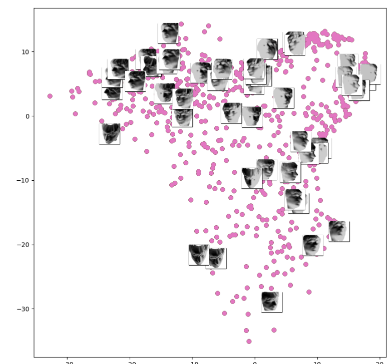
$\epsilon = 12$



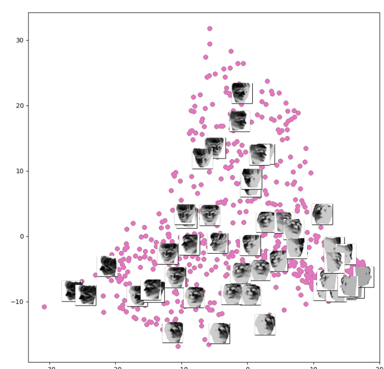
$\epsilon = 14$



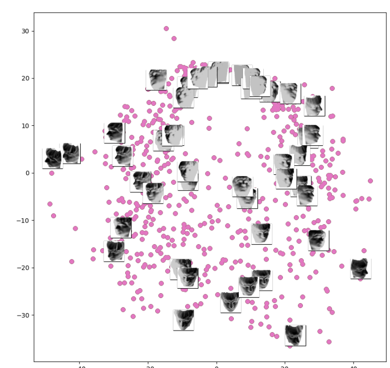
$\epsilon = 16$



$\epsilon = 18$

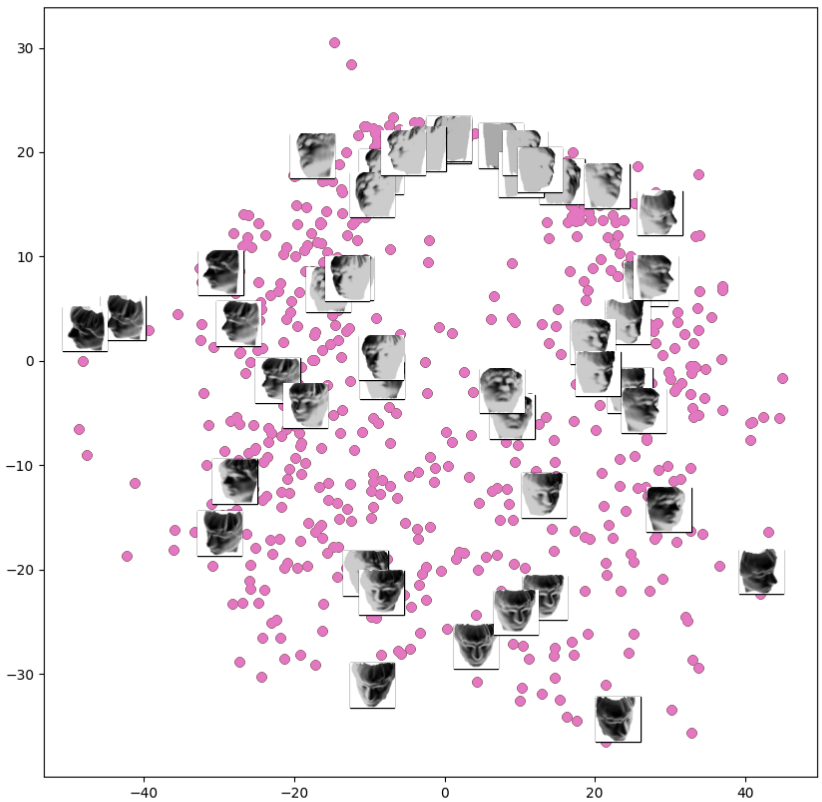


$\epsilon = 20$



$\epsilon = 22$

From the above inspection, the best ISOMAP is the one with $\epsilon = 12$, not 16 as I had done above.

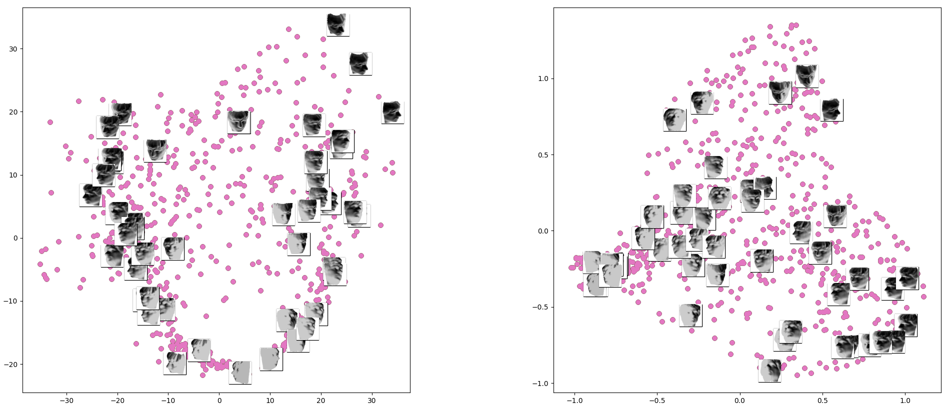


ISOMAP with $\epsilon = 12$

The scatterplot shows how the image faces do follow each other around the plot. The top of the image has the face looking down, the face looking to the left on the left side and right on the right side. As the images move around the plot, so does the direction of the face, appearing as it is a person doing a head roll around the plot. It could be that the placement of the shadow in the face is driving the similarity, as the right and top sides of the head are in the most shadow.

I find it interesting that in the plot, the faces on the right are looking to the right, and the faces on the left are looking to the left. This is true of the other tuning parameters as well.

3. Perform PCA on the images and project them into the top 2 principal components, and show them on a scatter plot.



ISOMAP ϵ = PCA

ISOMAP has a more meaningful projection than PCA. The PCA map is inconsistent with faces pointing in different directions. The PCA graph appears to be more diveded by color, with the left of the plot having less shadow than the right side. On the left side, the images are facing both left and right, although both are in profile and light in color. This is in contrast to the ISOMAP projection, where the direction of the face is aligned, not just the color or amount of shadow.

Eigenfaces and Simple Face Recognition

This question is a simplified illustration of using PCA for face recognition. We will use a subset of data from the famous Yale Face dataset.

Remark: Perform downsampling of the image by a factor of 4 to turn them into a lower resolution image as a preprocessing (e.g., reduce a picture of size 16-by-16 to 4-by-4).

First, given a set of images for each person, we generate the eigenface using these images.

1. Perform analysis on the Yale face dataset for Subject 1 and Subject 2, respectively, using all the images EXCEPT for the two pictures named subject01-test.gif and subject02-test.gif. **Plot the first 6 eigenfaces for each subject.**

   In my analysis, I first did subject one and then subject 2. The following are the steps I took for one subject:

   (a) Create a numpy array of the vectorized images.

   (b) Calculate the mean center, X, for the numpy arraw.

   (c) Find the U, Σ, and T, per question 2, for the X matrix.

   (d) For each of the first six eiegenfaces, reshape and create a new image.

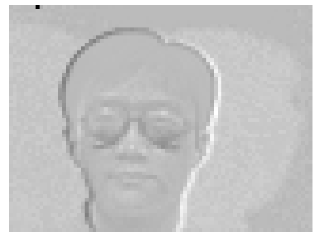
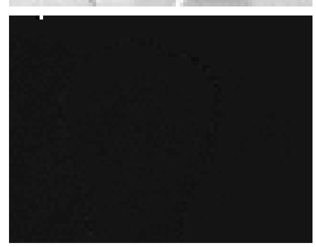
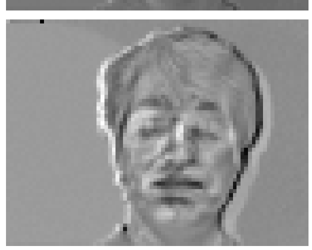
   After following the steps, the images on the next page are returned.

   What is interesting at firt is how diferent the two Subjects are from one another. The first components chosen for subject 1 are light colored, while for subject two they are closer to grey. While the colors are different, the expressions are both neutral.

   Similarly, the second eigenface shows similar expressions, a smile, while they have different colors, again with subject 2 much lighter. Interestingly, the glasses are not a large part of the image.

   As the images progress, they are showing different levels of brightness. Each image has a starkly different main color, going from almost white to almost black. That could tell me that the components are connected through color.

   The expressions of the faces remain either neutral or as a smile, they are not divided like the colors. This could mean that the expression is not a main deciding factor, leading to an "average" mouth which is neutral to smiling.

eigenface	**Subject 1**	**Subject 2**
original		
1		
2		
3		
4		
5		
6		

2. Now we will perform a simple face recognition task.

Face recognition through PCA is proceeded as follows. Given the test image subject01-test.gif and subject02-test.gif, first downsize by a factor of 4 (as before), and vectorize each image. Take the top eigenfaces of Subject 1 and Subject 2, respectively. Then we calculate the *projection residual* of the 2 vectorized test images with the vectorized eigenfaces:

$$s_{ij} = \|(\text{test image})_j - (\text{eigenface}_i)(\text{eigenface})_i^T(\text{test image})_j\|_2^2$$

Report all four scores: s_{ij}, $i = 1, 2$, $j = 1, 2$.

First I created a reduced vector of each test image and subtracted its mean. From this value, I input the above equation to test the image.

The test was run on multiple components to see if there is an optimal number of components for face recognition.

PC's	$s_{1,1}$	$s_{2,1}$	S1 Correct	$s_{2,2}$	$s_{1,2}$	S2 Correct
1	30805641	34716020	Yes	24357243	23102451	No
2	32006634	32938313	Yes	29800564	28193454	No
3	32315234	33637741	Yes	29317829	29507685	Yes
4	32615887	33993706	Yes	30440191	30506183	Yes
5	32566797	34273402	Yes	31429445	31107151	No
6	32666958	34473786	Yes	32023531	31472330	No

The table above summarizes the findings.

Subject 1 is correctly identified when $s_{1,1} < s_{2,1}$. The inequality tells us that our test image for subject 1 was closer to the compiled subject 1 than subject 2. For subject 2, the inequality is $s_{2,2} < s_{1,2}$

For this test, Subject 1 is correctly identified for all of the first 6 components. Subject 2, however, is only correctly identified with 3 or 4 components, which tells me that the subject 2 face is harder to identify. This could be because the picture of subject 1 has greater changes in the shadow, while the subject 2 pictures is flat.

The face algorithm works ok for Subject 1, but it does not work well with Subject 2. Subject 2 was identified correctly less than half the time, which is worse than if it were picked at random.

For this algorithm we used PCA, which usually works well for 2D mapping like seen in this exercise. Testing it with ISOMAP may not give a better result, but I would like to try.

To improve results, a more robust image database could be used. We could also not compress the images. Both of these techniques would, however, be computationally expensive.