ISYE 6740 Spring 2022
Homework 2
(100 points + 7 bonus points)

1. Conceptual questions [15 points].

1. (5 points) Please prove the first principle component direction v corresponds to the largest eigenvector of the sample covariance matrix.

$$v = \arg\max_{w:||w||\leq 1} \frac{1}{m}\sum_{i=1}^{m}(w^T x^i - w^T \mu)^2$$

You may use the proof steps in the lecture, but you should represent them logically and cohesively.

2. (5 points) What is the relationship between SVD and eigendecomposition? Please explain this mathematically, and touch on why this is relevant for PCA.

3. (5 points) Explain the three key ideas in ISOMAP (for manifold learning and non-linear dimensionality reduction).

2. Eigenfaces and simple face recognition [42 points; including 2 bonus points.]

This question is a simplified illustration of using PCA for face recognition, via the eigenfaces method. We will use a subset of data from the famous Yale Face dataset. If you are interested in reading further into this approach, see 'Face recognition using eigenfaces, M.A Turk and A.P. Pentland, IEEE computer society conference on computer vision and pattern recognition (1991) 586-587'.

Remark: You will have to 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). Also, while this method does not explicitly require PCA, similar preprocessing steps (mean-centering) should be considered. In this question, you can implement your own code or call packages.

First, given a set of images for each person, we generate the eigenface using these images. You will treat one picture from the same person as one data point for that person. Note that you will first vectorize each image, which was originally a matrix. Thus, the data matrix (for each person) is a matrix; each row is a vectorized picture. You will find weight vectors to combine the pictures to extract different "eigenfaces" that correspond to that person's pictures' first few principal components.

(a) (20 points) 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.** When visualizing, please reshape the eigenvectors into proper images. Please explain can you see any patterns in the top 6 eigenfaces?

(b) (20 points) 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}, $i = 1, 2$, $j = 1, 2$. Explain how to recognize the faces of the test images using these scores.

Note: Pay close attention to the shapes of your vectors/matrices when performing this calculation.

(c) (Bonus: 2 points) Explain if eigenface facial recognition can work well, and discuss how it can be improved.

3. Order of faces using ISOMAP [45 points]

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 that we have also seen in the lecture as an exercise (isn't this exciting to go through the process of generating results for a high-impact research paper!)

The file isomap.mat (or isomap.dat) 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. (This is one of the datasets used in the original paper.) In this question, you are expected to implement the ISOMAP algorithm by coding it up yourself. You may find the shortest path (required by one step of the algorithm), using `https://scikit-learn.org/stable/modules/generated/sklearn.utils.graph_shortest_path.graph_shortest_path.html`.

Using Euclidean distance (i.e., in this case, a distance in $\mathbb{R}^{4096}$) to construct the ϵ-ISOMAP (follow the instructions in the slides.) You will tune the ϵ parameter to achieve the most reasonable performance. Please note that this is different from K-ISOMAP, where each node has exactly K nearest neighbors.

(a) (10 points) Visualize the nearest neighbor graph (you can either show the adjacency matrix (e.g., as an image), or visualize the graph similar to the lecture slides using graph visualization packages such as Gephi (`https://gephi.org`) and illustrate a few images corresponds to nodes at different parts of the graph, e.g., mark them by hand or use software packages).

(b) (15 points) Implement the ISOMAP algorithm yourself to obtain a two-dimensional low-dimensional embedding. Plot the embeddings using a scatter plot, similar to the plots in lecture slides. Find a few images in the embedding space and show what these images look like and specify the face locations on the scatter plot. Comment on do you see any visual similarity among them and their arrangement, similar to what you seen in the paper?

(c) (10 points) Now choose ℓ_1 distance (or Manhattan distance) between images (recall the definition from "Clustering" lecture)). Repeat the steps above. Use ϵ-ISOMAP to obtain a $k = 2$ dimensional embedding. Present a plot of this embedding and specify the face locations on the scatter plot. Do you see any difference by choosing a different similarity measure by comparing results in Part (b) and Part (c)?

(d) (10 points) Perform PCA (you can now use your implementation written in Question 1) on the images and project them into the top 2 principal components. Again show them on a scatter plot. Explain whether or you see a more meaningful projection using ISOMAP than PCA.

(e) (No credit, but a invitation to reflect): What did you think about reproducing this paper (and the eigenface paper). What did you enjoy? Were there any challenges, and if so, what were they?

4. PCA: Food consumption in European countries [Bonus Question: 5 points]

The data food-consumption.csv contains 16 countries in Europe and their consumption for 20 food items, such as tea, jam, coffee, yogurt, and others. We will perform principal component analysis to explore the data. In this question, please implement PCA by writing your own code (you can use any basic packages, such as numerical linear algebra, reading data, in your file).

First, we will perform PCA analysis on the data by treating each country's food consumption as their "feature" vectors. In other words, we will find weight vectors to combine 20 food-item consumptions for each country.

(a) (3 points) For this problem of performing PCA on countries by treating each country's food consumption as their "feature" vectors, explain how the data matrix is set-up in this case (e.g., the columns and the rows of the matrix correspond to what). Now extract the first two principal components for each data point (thus, this means we will represent each data point using a two-dimensional vector). Draw a scatter plot of two-dimensional representations of the countries using their two principal components. Mark the countries on the lot (you can do this by hand if you want). Please explain any pattern you observe in the scatter plot.

(c) (2 points) Now, we will perform PCA analysis on the data by treating country consumptions as "feature" vectors for each food item. In other words, we will now find weight vectors to combine country consumptions for each food item to perform PCA another way. Project data to obtain their two principle components (thus, again each data point – for each food item – can be represented using a two-dimensional vector). Draw a scatter plot of food items. Mark the food items on the plot (you can do this by hand if you do not want). Please explain any pattern you observe in the scatter plot.