

ISYE 6740 Spring 2025  
Homework 2  
(100 points + 5 bonus points)

**Provided Data:**

- Q2: PCA: Food Consumption [food-consumption.csv]
- Q3: Order of faces using ISOMAP [isomap.dat, isomap.mat]
- Q5: Eigenfaces and simple face recognition [yalefaces]

**All results that are present only in code/notebooks, but not in your PDF report will not be accepted for points.**

**For any code that requires randomness, please set your seed as 6740**

**1. Conceptual questions [30 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 please write them logically and cohesively.

2. (5 points) Based on your answer to the question above, explain how to further find the second and third largest principle component directions.
3. (5 points) Consider the diagonal matrix  $A = \begin{bmatrix} 3 & 0 \\ 0 & 2 \end{bmatrix}$ . Find two distinct eigenvalue decompositions of  $A$ , and prove mathematically that both are valid. This demonstrates that eigendecomposition is not unique. **Note:** Your eigenvalue decompositions must be done mathematically, not programatically.
4. (5 points) Explain the three key ideas in ISOMAP (for manifold learning and non-linear dimensionality reduction).
5. (5 points) Explain how to decide  $k$ , the number of principle components, from data.
6. (5 points) How do outliers affect the performance of PCA? You can create numerical examples to study and show this.

## 2. PCA: Food consumption in European countries [20 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.

1. (10 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 plot (you can do this by hand if you want). Please explain any pattern you observe in the scatter plot.
2. (10 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 want). Please explain any pattern you observe in the scatter plot.

### 3. Order of faces using ISOMAP [25 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  $64 \times 64$  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://docs.scipy.org/doc/scipy/reference/generated/scipy.sparse.csgraph.shortest\\_path.html](https://docs.scipy.org/doc/scipy/reference/generated/scipy.sparse.csgraph.shortest_path.html).

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

1. (5 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).
2. (10 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? Come up with a way to tune the kernel bandwidth to have desired result.
3. (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.

#### 4. Eigenfaces and simple face recognition [25 points].

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:** 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). 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.

1. (10 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?

2. (10 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.

3. (5 points) Comment if your face recognition algorithm works well and discuss how you would like to improve it if possible.

**5. To subtract or not to subtract, that is the question [Bonus: 5 points].**

In PCA, we have to subtract the mean to form the covariance matrix

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

before finding the weight vectors, where  $\mu = \frac{1}{m} \sum_{i=1}^m x^i$ . For instance, we let

$$Cw^1 = \lambda_1 w^1$$

where  $\lambda_1$  is the largest eigenvalue of  $C$ , and  $w^1$  is the corresponding largest eigenvector.

Now suppose Prof. X insisting not subtracting the mean, and uses the eigenvectors of

$$\tilde{C} = \frac{1}{m} \sum_{i=1}^m x^i x^{iT}$$

to form the weight vectors. For instance, she lets  $\tilde{w}^1$  to be such that

$$\tilde{C}\tilde{w}^1 = \tilde{\lambda}_1 \tilde{w}^1$$

where  $\tilde{\lambda}_1$  is the largest eigenvalue of  $\tilde{C}$ .

Now the question is, are they the same (with and without subtract the mean)? Is  $w^1$  equal or not equal to  $\tilde{w}^1$ ? Use mathematical argument to justify your answer.