ISYE 6740 Fall 2024
Homework 2
(100 points + 5 bonus points)

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.

Answer:

$$\begin{aligned}\frac{1}{m}\sum_{i=1}^{m}\left(w^\top x^i - w^\top \mu\right)^2 &= \frac{1}{m}\sum_{i=1}^{m}\left(w^\top\left(x^i-\mu\right)\right)^2 \\ &= \frac{1}{m}\sum_{i=1}^{m} w^\top\left(x^i-\mu\right)\left(x^i-\mu\right)^\top w \\ &= w^\top\left(\frac{1}{m}\sum_{i=1}^{m}\left(x^i-\mu\right)\left(x^i-\mu\right)^\top\right)w \\ &= w^\top C w\end{aligned}$$

C is the covariate matrix, w is the direction matrix Lagrangian function of the constrained timization problem:

$$L\left(w,\lambda\right) = w^\top C w + \lambda\left(1-\|w\|^2\right)$$

In order to caluclate the maximum w, get the eigenvalue and eigenvector pair:

$$\frac{\partial L}{\partial w} = 0 = 2Cw - 2\lambda w \iff Cw = \lambda w$$

The optimal solution w is an eigenvector of C and λ is an eigenvalue,

$$w^\top C w = \lambda w^\top w = \lambda\|w\|^2$$

The direction that maximizes the variance corresponds to v with the largest eigenvalue λ. Therefore, the first principle component direction v is the eigenvector of the sample covariance matrix with largest eigenvalue.

2. (5 points) Based on your answer to the question above, explain how to further find the second largest principle component directions.

Answer:

$$C = W\Lambda W^T$$

W: w^1, w^2,..., w^n (called the eigenvectors)
Λ: $diag(\lambda_1, \lambda_2, ..., \lambda_n)$ (called the eigenvalues)

To find the second principal component C_2 , we need to subtract the influence of the first principal component from the covariance matrix.

$$C_2 = C - w\lambda w^T$$

Then, the second largest principle component directions.

$$w_2 = \arg \max_{w_2 : \|w_2\| \leq 1} w^T C_2 w$$

3. (5 points) Based on the outline given in the lecture, show that the maximum likelihood estimate (MLE) for Gaussian random variable using observations $x^1, \ldots, x^m$, that are *i.i.d.* (independent and identically distributed) following the distribution $\mathcal{N}(\mu, \sigma^2)$, and the mean and variance parameters are given by

$$\hat{\mu} = \frac{1}{m}\sum_{i=1}^{m} x^i, \quad \hat{\sigma}^2 = \frac{1}{m}\sum_{i=1}^{m}(x^i - \hat{\mu})^2,$$

respectively. Please show the work for your derivations in full detail.

Answer:
The pdf of Gaussian distributin is:

$$f(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

Since the data points are *i.i.d.*, the likelihood function $L(\mu, \sigma^2)$ is:

$$\begin{aligned} L(\mu, \sigma^2) &= \prod_{i=1}^{m} f(x^i | \mu, \sigma^2) \\ &= \prod_{i=1}^{m} \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \end{aligned}$$

Hence the log-likelihood is:

$$\log L(\mu, \sigma^2) = m \log \frac{1}{\sqrt{2\pi\sigma^2}} - \frac{1}{2\sigma^2}\sum_{i=1}^{m}(x_i - \mu)^2$$

Take the derivative of $\log L(\mu, \sigma^2)$ with respect to μ and let it equals to 0:

$$\frac{\partial \ell(\mu, \sigma^2)}{\partial \mu} = \frac{1}{\sigma^2}\sum_{i=1}^{m}(x^i - \mu) = 0$$

$$\sum_{i=1}^{m}(x^i - \mu) = 0$$

$$\mu = \frac{1}{m}\sum_{i=1}^{m} x^i$$

The MLE for μ is:

$$\hat{\mu} = \frac{1}{m}\sum_{i=1}^{m} x^i$$

Take the derivative of $\log L(\mu, \sigma^2)$ with respect to σ^2 and let it equals to 0:

$$\frac{\partial \ell(\mu, \sigma^2)}{\partial \sigma^2} = -\frac{m}{2\sigma^2} + \frac{1}{2(\sigma^2)^2}\sum_{i=1}^{m}(x^i - \mu)^2 = 0$$

The MLE for σ^2 is:

$$\hat{\sigma}^2 = \frac{1}{m}\sum_{i=1}^{m}(x^i - \mu)^2$$

4. (5 points) Explain the three key ideas in ISOMAP (for manifold learning and non-linear dimensionality reduction).

   Answer:
   1. There are two types of ISOMAP, $\epsilon - ISOMAP$ and $k - ISOMPA$. For $k - ISOMPA$ select k nearest neighbors for each node, for $k - ISOMAP$ it can be performed by connecting each point to other points within fixed radius ϵ, it's also considered a weighted nearest neighboor.
   2. ISOMAP constructs a nearest-neighbor graph based on Euclidean distances in order to get the geodesic distance. Find the shortest path distance matrix D between all pairs of points, also called graph distance matrix, which can be computed with Floyd-Warshall algorithm and $m(m-1)/2$ applications of Dijkstra's algorithm
   3. ISOMAP can use Multi-dimensional Scaling (MDS) algorithm to give pairwise dissimilarity between data points, reconstruct a low-dimensional "map" that preserves distances.

5. (5 points) Explain how to decide k, the number of principle components, from data.

   Answer:
   The eigenvalue λ_i represents the variations which are "signals" or information in the data. The components with the lowest eigenvalues contain the least information, so they can be dropped. The importance of each component is represented by explained variance ratio, which indicates the portion of the variance of each principal component.

   Cumulative explained variance is the summation of the explained variance ratio of the first k components.

   Typically we choose the k for which the cumulative explained variance exceeds 95%.

   $$\text{Cumulative explained variance of the first } k \text{ components} = \frac{\sum_{i=1}^{k}\lambda_i}{\sum_{j=1}^{D}\lambda_j} > 95\%$$

6. (5 points) How do outliers affect the performance of PCA? You can create numerical examples to study and show this.

   Answer: Outliers significantly affect the performance of PCA because PCA is sensitive to the variance in the data. Since outliers tend to have a disproportionately large influence on the covariance matrix, they can distort the directions of the principal components.

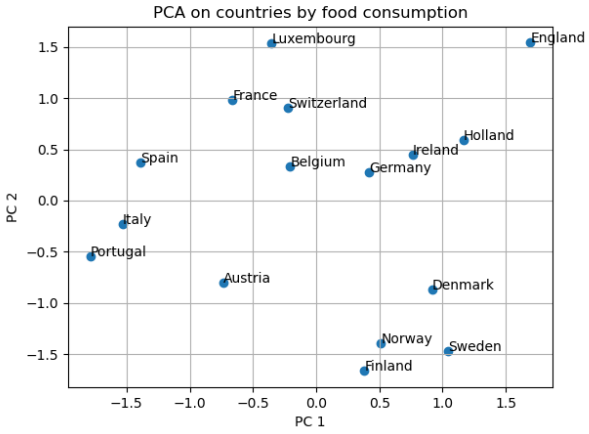
2. PCA: Food consumption in European countries [20 points].

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.

   Answer:

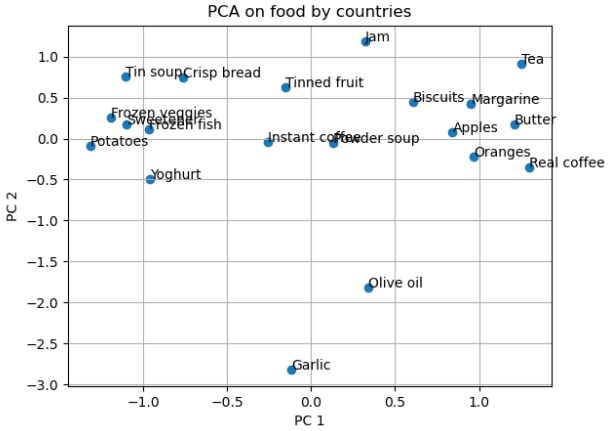


   Denmark, Sweden, Norway, and Finland are clustered together, indicating similar food consumption patterns. England stands out on the right of the plot, suggesting that its food consumption patterns are quite different from other countries especially Portugal.

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.

   Answer:

   Tea, Real coffee, Margarine, Oranges, Butter are clustered on the right, indicating that these food items are consumed in similar patterns across countries. Olive oil and Garlic are positioned distinctly on the lower left, indicating that their consumption patterns are quite unique compared to most other food items.


PCA on food by countries
Jam
Tea
Tin soup
Crisp bread
Tinned fruit
Biscuits
Margarine
Frozen veggies
Butter
Frozen fish
Apples
Potatoes
Instant coffee
Powder soup
Oranges
Real coffee
Yoghurt
Olive oil
Garlic
PC 1
PC 2
1.0
0.5
0.0
−0.5
−1.0
−1.5
−2.0
−2.5
−3.0
−1.0
−0.5
0.0
0.5
1.0

3. Order of faces using ISOMAP [25 points]

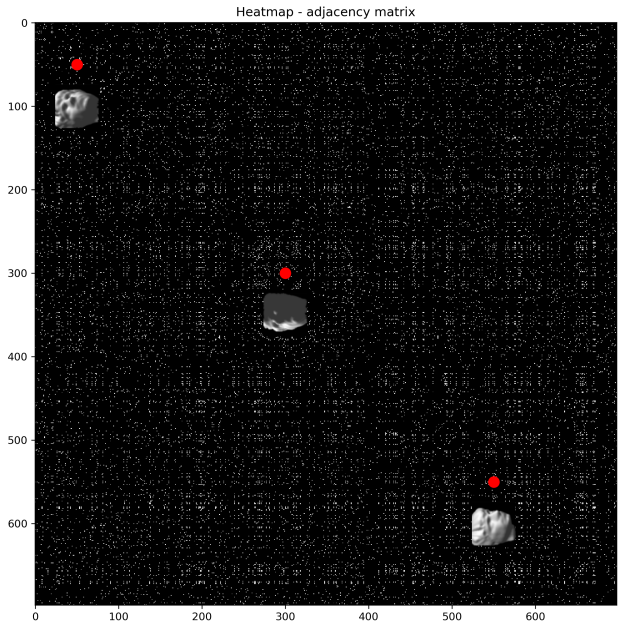
This question aims to reproduce the ISOMAP algorithm results in the original paper for ISOMAP, J.B. Tenenbaum, V. de Silva, and J.C. Langford, Science 290 (2000) 2319-2323 that we have also seen in the lecture as an exercise (isn't this exciting to go through the process of generating results for a high-impact 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 64×64 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.

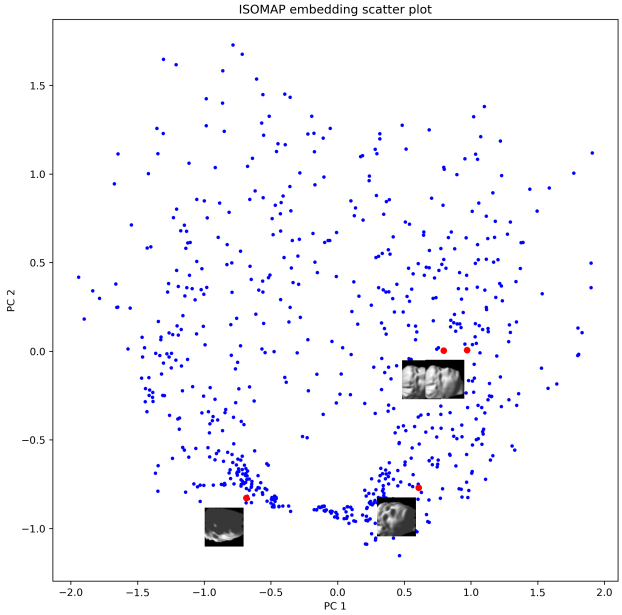
1. (5 points) Visualize the nearest neighbor graph (you can either show the adjacency matrix (e.g., as an image), or visualize the graph similar to the lecture slides using graph visualization packages such as Gephi (https://gephi.org) and illustrate a few images corresponds to nodes at different parts of the graph, e.g., mark them by hand or use software packages).

   Answer:



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?

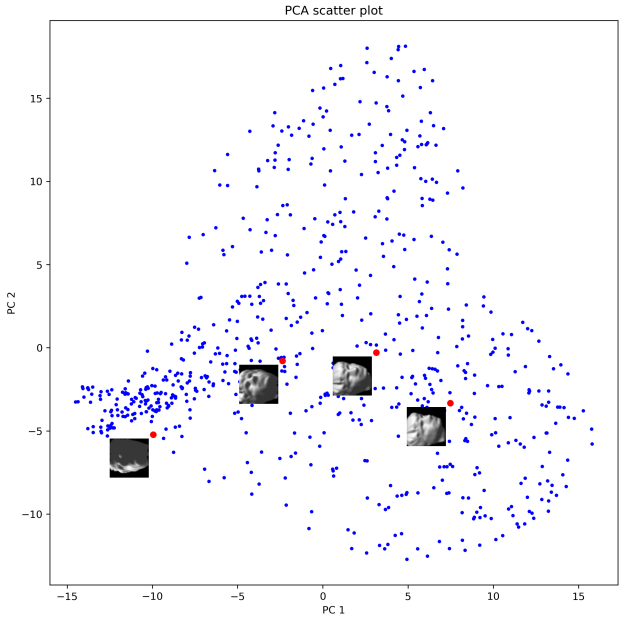
   Answer:



We can see the faces in the upper right show similar pattern. This suggests that ISOMAP can effectively separate different variations in the dataset (such as facial poses or expressions).

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.

   Answer:



The images are placed at various positions across the scatter plot, but the clusters are less distinct

than in the ISOMAP plot. PCA doesn't seem to cluster the images in a good way when compared with ISOMAP.

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.

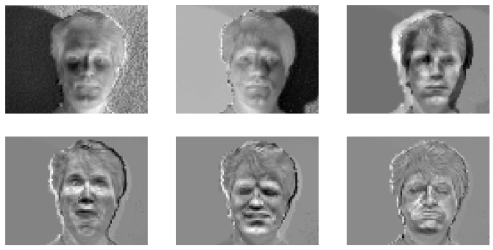
Remark: You will have to perform downsampling of the image by a factor of 4 to turn them into a lower resolution image as a preprocessing (e.g., reduce a picture of size 16-by-16 to 4-by-4). In this question, you can implement your own code or call packages.

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?

   Answer:

   For subject 1:

   For subject 2:

   The top 3 images correspond to the first three principal components (eigenfaces) which usually captures the most significant variance.

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.

Answer:

s11 7.2016e+06
s12 4.0638e+07
s21 3.3747e+07
s22 4.2425e+06

We can get s11 less than s21 and s22 less than s12, which means Test Image 1 is closer to Subject 1, and Test Image 2 is closer to Subject 2.

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

Answer:

The model works good on this dataset. For this kind of small dataset the algorithm is likely to provide decent results, especially when there are clear differences between subjects. There are also some improvements: Augment the training data with more diverse examples and preprocess the images to reduce noise and normalize lighting conditions, ensuring better recognition accuracy.

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 λ_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.