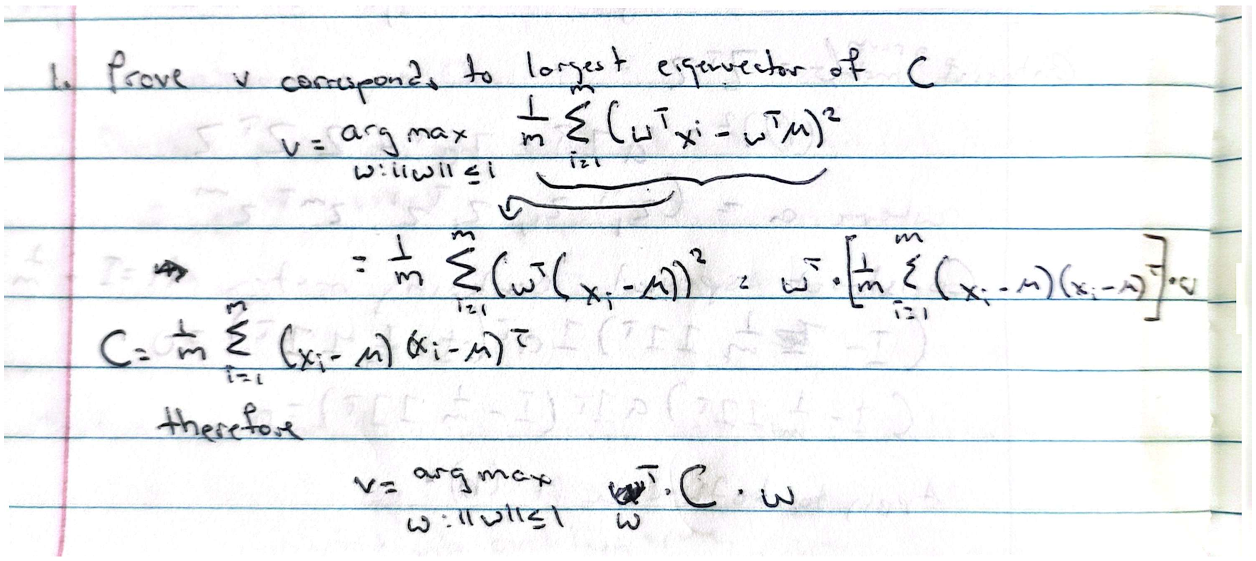
ISYE 6740 HW 2

Mohammed Khalid Siddiqui

September 16, 2024

1 Conceptual Questions

1. Proving the first principle component direction v corresponds to the largest eigenvector of the sample covariance matrix provided below:



1. Prove v corresponds to largest eigenvector of C

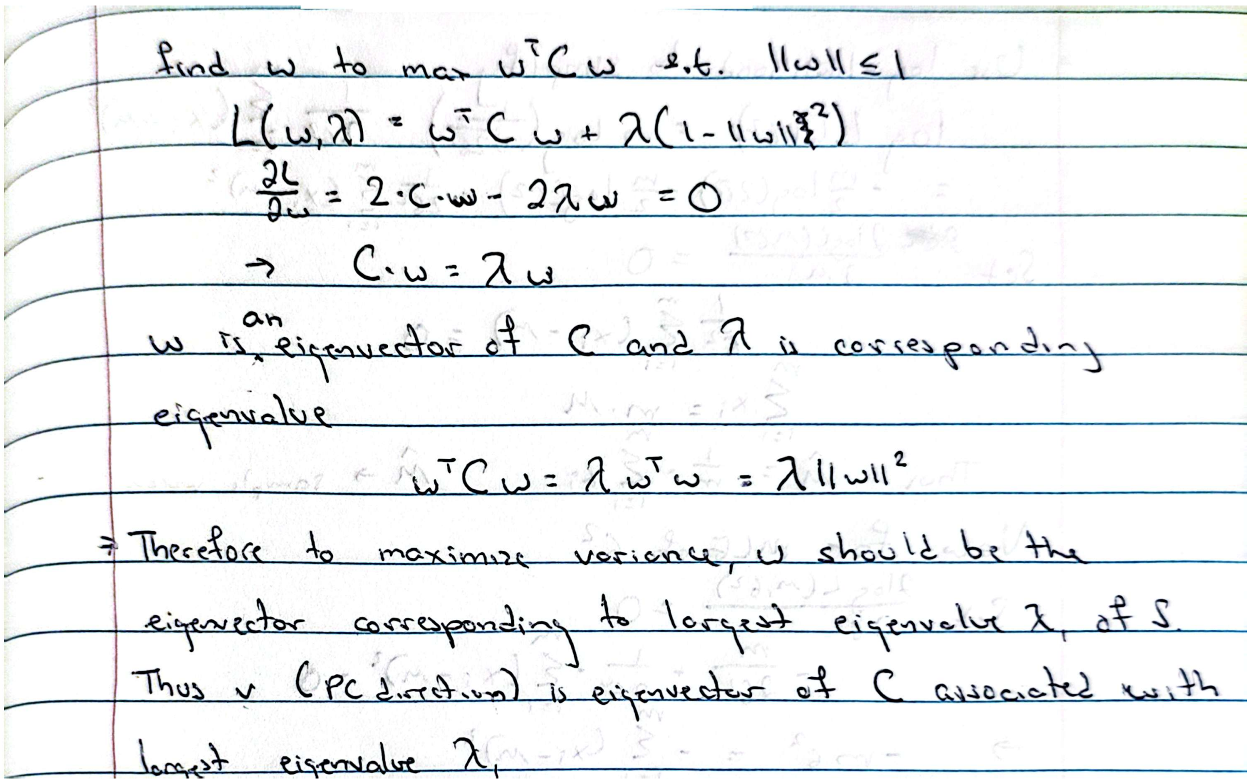
$$v = \underset{w:\|w\| \le 1}{\arg\max} \; \underbrace{\frac{1}{m}\sum_{i=1}^{m}(w^T x^i - w^T \mu)^2}$$

$$= \frac{1}{m}\sum_{i=1}^{m}(w^T(x_i - \mu))^2 = w^T \cdot \left[\frac{1}{m}\sum_{i=1}^{m}(x_i - \mu)(x_i - \mu)^T\right] \cdot w$$

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

therefore

$$v = \underset{w:\|w\| \le 1}{\arg\max} \; w^T \cdot C \cdot w$$



find w to max $w^T C w$ s.t. $\|w\| \le 1$

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

$$\frac{\partial L}{\partial w} = 2 \cdot C \cdot w - 2\lambda w = 0$$

$$\rightarrow \quad C \cdot w = \lambda w$$

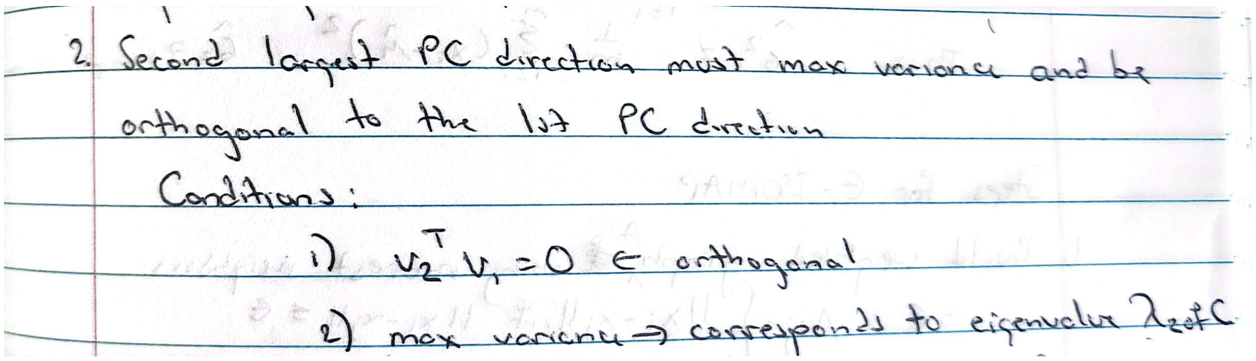
w is $\overset{an}{}$ eigenvector of C and λ is corresponding eigenvalue

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

$\Rightarrow$ Therefore to maximize variance, w should be the eigenvector corresponding to largest eigenvalue λ_1 of S.

Thus v (PC direction) is eigenvector of C associated with largest eigenvalue λ_1

2. Extend above logically with criteria to find the 2nd principle component direction v below:



2. Second largest PC direction must max variance and be orthogonal to the 1st PC direction

Conditions:

1) $v_2^T v_1 = 0 \leftarrow$ orthogonal

2) max variance $\Rightarrow$ corresponds to eigenvalue λ_2 of C

3. Show MLE for Gaussian RV are given by the below:

3. Show MLE for Gaussian RV $_{\wedge}^{\text{using obs}}$ $x_1 \ldots x_m$ iid following dist. $N(\mu, \sigma^2)$ and mean & var. params are given by

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

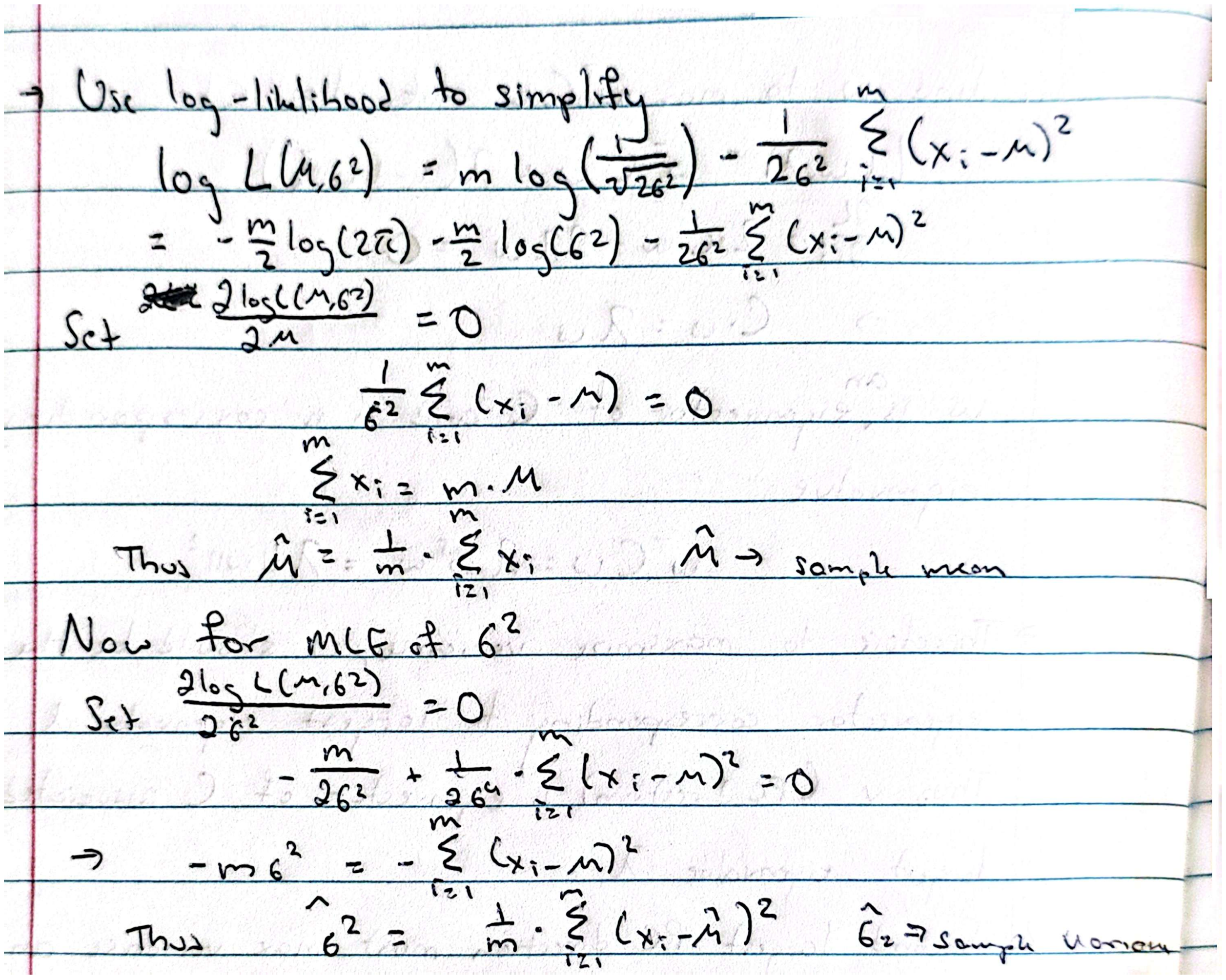
pdf $N(\mu, \sigma^2)$

$$\hookrightarrow p(x_i \mid \mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x_i-\mu)^2}{2\sigma^2}\right)$$

Likelihood func. L

$$L(\mu, \sigma^2) = \prod_{i=1}^{m} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x_i-\mu)^2}{2\sigma^2}\right) \rightarrow \prod_{i=1}^{m} p(x_i \mid \mu, \sigma^2)$$

$$= \left(\frac{1}{\sqrt{2\pi\sigma^2}}\right)^m \exp\left(-\frac{1}{2\sigma^2}\sum_{i=1}^{m}(x_i-\mu)^2\right)$$



→ Use log-likelihood to simplify

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

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

Set $\frac{\partial \log L(\mu,\sigma^2)}{\partial \mu} = 0$

$$\frac{1}{\sigma^2}\sum_{i=1}^{m}(x_i-\mu) = 0$$

$$\sum_{i=1}^{m} x_i = m\cdot\mu$$

Thus $\hat{\mu} = \frac{1}{m}\cdot\sum_{i=1}^{m} x_i$ $\hat{\mu} \rightarrow$ sample mean

Now for MLE of σ^2

Set $\frac{\partial \log L(\mu,\sigma^2)}{\partial \sigma^2} = 0$

$$-\frac{m}{2\sigma^2} + \frac{1}{2\sigma^4}\cdot\sum_{i=1}^{m}(x_i-\mu)^2 = 0$$

→ $-m\sigma^2 = -\sum_{i=1}^{m}(x_i-\mu)^2$

Thus $\hat{\sigma}^2 = \frac{1}{m}\cdot\sum_{i=1}^{m}(x_i-\hat{\mu})^2$ $\hat{\sigma}^2 \rightarrow$ sample varian

4. There are three key ideas in ISOMAP (for manifold learning and non-linear dimensionality reduction).

   (a) Construct a neighborhood graph to capture local relationships and the geometry of the manifold. To build this graph, each point is connected to its nearest neighbors. The first method is to employ the Epsilon-ISOMAP (implemented in section 3) which uses a threshold Epsilon and returns all pairwise distances within that threshold and 0 if there are none. The second method is to select the K nearest neighbors for each point in what is called K-ISOMAP. The edges in this graph are weighted by the local Euclidean distance.

   (b) Next we compute geodesic distances by finding shortest path between all pairs of points in the manifold. To do so, you can employ Floyd-Warshall algorithm or Djikstra's and construct matrix D (the shortest path distance matrix). The algorithms above provide a pretty useful approximation of the true distance between points along the manifold. Through a combination of the neighborhood graph adjacency matrix and the geodesic distance computation with one of the algorithms above, we're able to overcome the limitations of linear methods like PCA or SVD.

   (c) Lastly, embed the data into a lower dimensional space while preserving the geodesic distances as much as possible. You can use Multi-Dimension Scaling (MDS) to do so and thereby convert the geodesic distance matrix into a form which can be used to employ dimensionality reduction technique such as eigenvalue decomposition. Identify leading components and project your original points into the lower dimensional space.

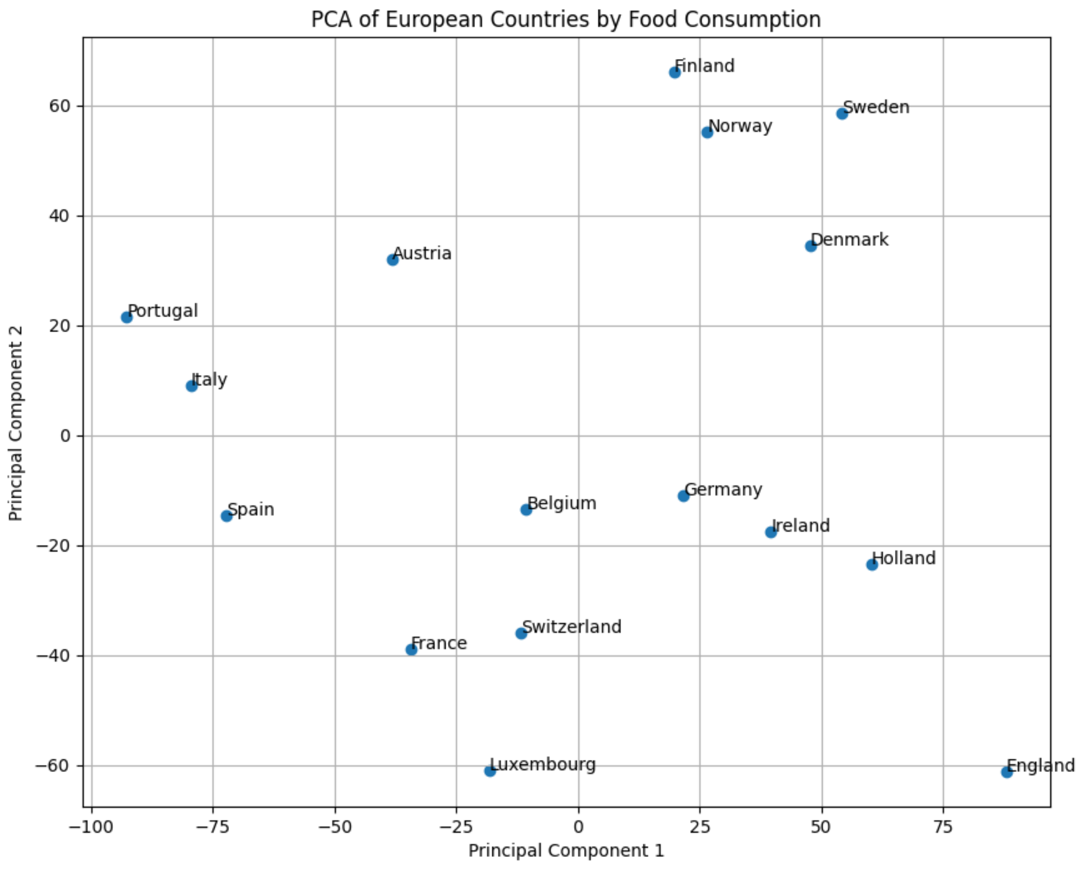
5. There are several ways you can decide the number of principal components k when performing PCA.
   (a) One commonly used method is the Explained variance criteria. One goal of PCA is to capture the variance in the original data. When performing PCA, the sum of the eigenvalues will represent the total variance of the data. Each eigenvalue represents how much of the total variance a Principal Component represents. You may set a threshold, such as 95%, and then calculate the cumulative sum of this variance, choosing the least number of principal components with the largest individual variances which sum up to meet or exceed the threshold you've set.
   (b) Another popular method is known as the "elbow" method or through the use of a Scree plot. You simply plot the eigenvalues in descending order with the principal component number as your x-axis. You visually determine the point where the rate of decrease of the eigenvalues slow down significantly, the "elbow". You'll then choose k at the elbow point. Note that the elbow may not always be clear.
   (c) Another method known as the Kaiser criterion suggests simply choosing eigenvalues greater than 1 though this may be too conservative and not generalize well to high dimensional data with a lot of really small eigenvalues.
   (d) Another tried and true method is employing the use of cross validation to choose the number of components k based on the performance of the model itself on training data. You will choose the model where the test error from your CV runs is the lowest. This may be computationally expensive due to the nature of CV.

6. PCA is sensitive to outliers because it relies on variance and the computation of C, the covariance matrix. To employ PCA, you first calculate the mean of the data and center the data around this mean. You then calculate the covariance matrix C. This captures how much the variables in the data vary together. You then apply eigen-decomposition to get the leading eigenvectors and eigenvalues of the covariance matrix to find the principal components that maximize the variance. Outliers can skew the direction of a principle component and thus lead to distorted or biased representations of the underlying data. It's therefore important to handle outliers before implementing PCA.

2 PCA: Food Consumption in European Countries

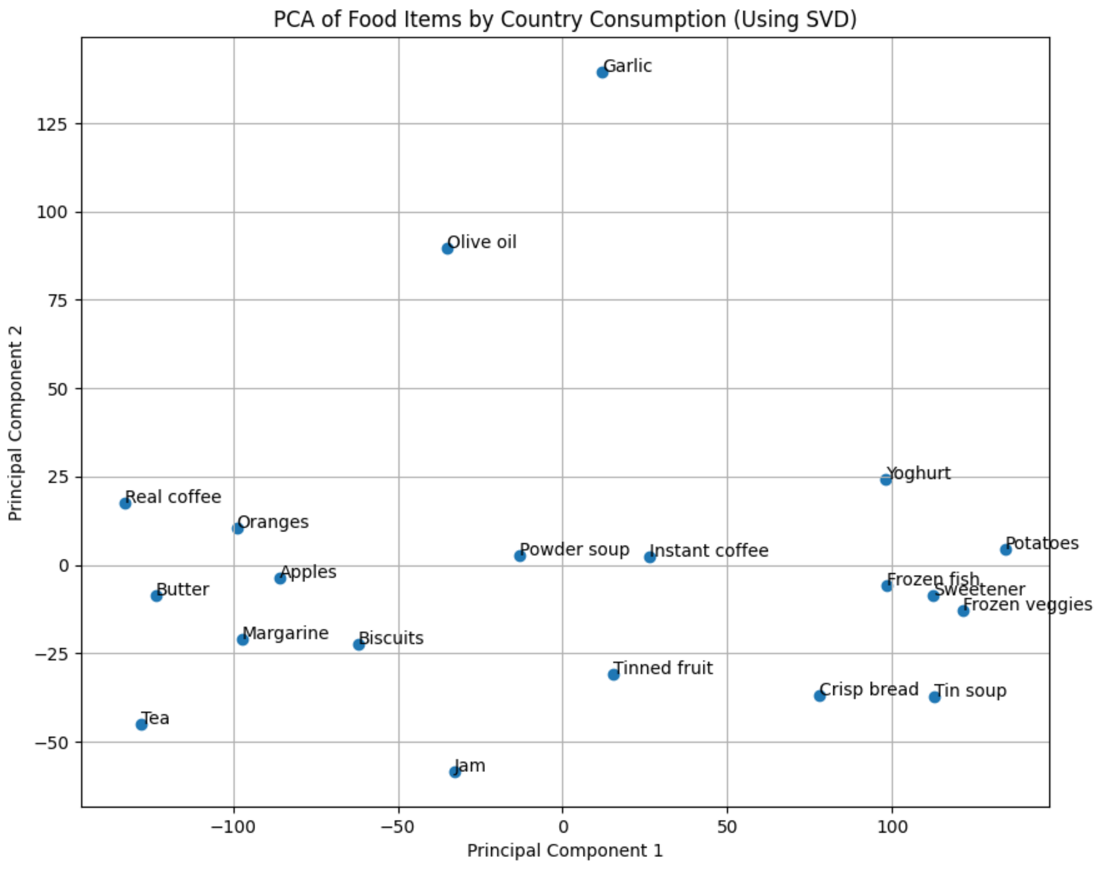
1. For this problem, we performed Principal Component Analysis (PCA) on 16 European countries based off of their food consumption habits represented by 20 columns of data. Each row represents a country and each column represents a food type. The data represents a numeric value on a consumption scale with 0 being none and 100 as the maximum. The idea behind implementing PCA is apply dimension reduction to represent distances between countries from each other in a visual representation as shown in the figure below. Despite reducing the number of dimensions for each country from 20 to 2, we can maintain distance relationships. For example, the consumption habits of the Scandinavian countries appear to be similar to each other, which is expected. The same applied for the western Mediterranean countries. Performing a visual sanity check, we can see that countries we expect to have similar food consumption habits do in fact exhibit these relationships in the lower dimensional representation.

   To attain this 2-dimensional representation for each country, we applied the following steps. First, we centered the data around the mean for each column so each cell now represented the distance from the mean where 0 implies the value of the cell is set to the mean of the column (food type) the cell is in. As our matrix is in the form m x n and not square, we used Singular Value Decomposition (SVD) instead of eigen-decomposition. After calculating the respective U, Sigma, and V-Transposed matrices, we then calculated the projected points in two dimensional space using the top two singular values multiplied with the respective values in the U matrices as per the recommendation from source 9 in the Sources section. Lastly, the first two principal components were plotted to attain the visual representation below.

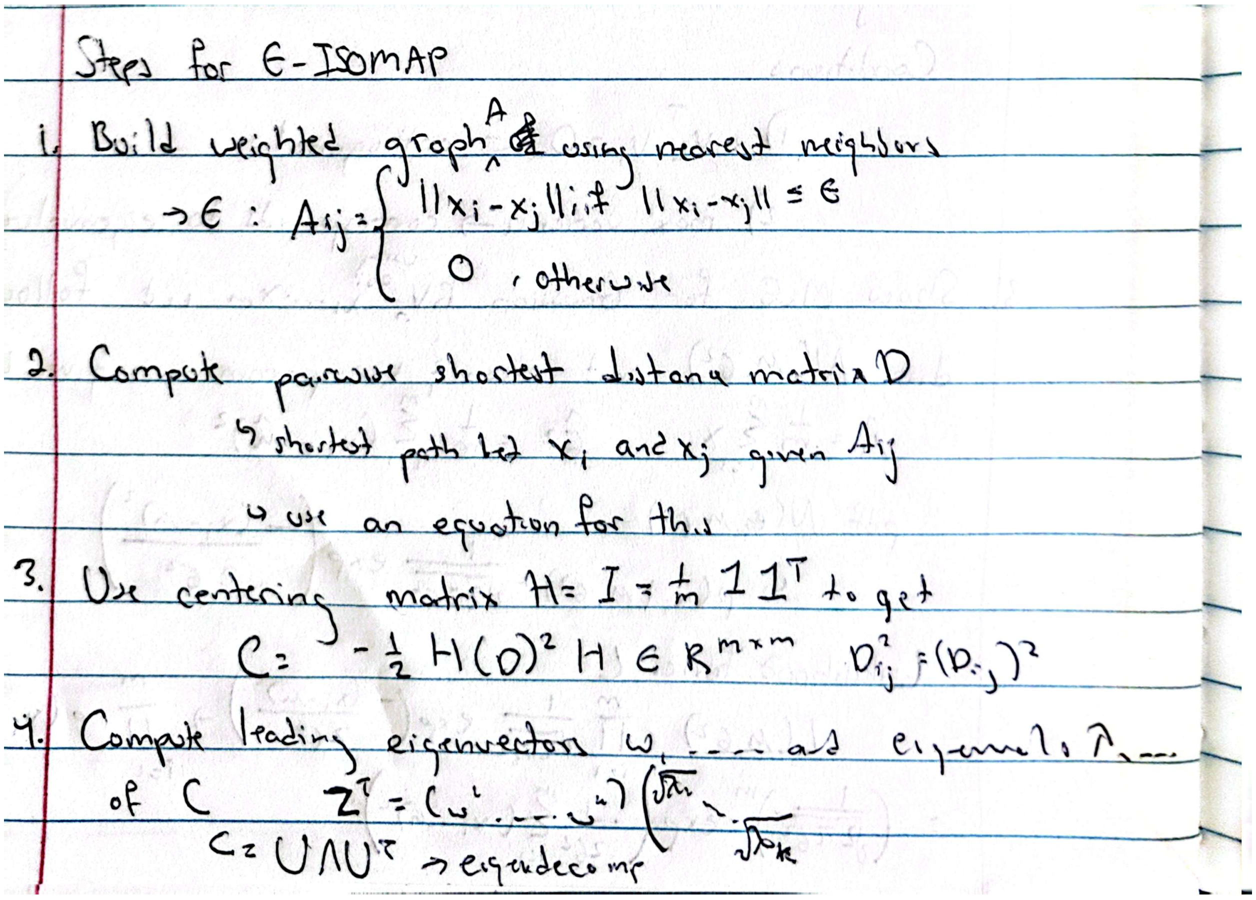


2. For this problem, we first transpose the original data frame so that our columns represent our 16 countries and our rows represent the 20 food items. We then go through the same steps above, first centering our column data around the mean, then performing SVD, and projecting our data back into 2 dimensions before visualing the top two principal components in the plot below. We can immediately see that foods which we expect to be eaten by similar countries seem to be clustered closer together. for example, we can impute that processed goods such as frozen vegetables, frozen fish, tin soup, and sweetener might show similar consumption patterns among the 16 countries, while apples, oranges, butter, and biscuits might exhibit similar consumption patterns to each other among countries.

   For both this and the previous visual, I didn't scale the projections as the relationships between the components seem to be maintained when I experimented with different scaling techniques in the projection step (such as multiplying the weight vector by the centered data and dividing by square root of lambda) and I found it a bit easier to interpret and explain the data without adding an additional scaling component while still preserving the relative relationships between components.


PCA of Food Items by Country Consumption (Using SVD)
Principal Component 1
Principal Component 2
Garlic
Olive oil
Yoghurt
Real coffee
Oranges
Powder soup
Instant coffee
Potatoes
Apples
Butter
Frozen fish
Sweetener
Frozen veggies
Margarine
Biscuits
Tinned fruit
Crisp bread
Tin soup
Tea
Jam
−100
−50
0
50
100
125
75
25
−25
−50

3 Order of Faces Using Epsilon-ISOMAP



Steps for ϵ-ISOMAP

1. Build weighted graph $\overset{A}{\wedge}$ ~~&~~ using nearest neighbors

$\to \epsilon$: $A_{ij} = \begin{cases} \|x_i - x_j\| \text{ if } \|x_i - x_j\| \le \epsilon \\ 0 \text{ , otherwise} \end{cases}$

2. Compute pairwise shortest distance matrix D

↳ shortest path bet x_i and x_j given A_{ij}

↳ use an equation for this

3. Use centering matrix $H = I = \frac{1}{m} \mathbb{1}\mathbb{1}^T$ to get

$C = -\frac{1}{2} H (D)^2 H \in \mathbb{R}^{m \times m}$ $\quad D_{ij}^2 = (D_{ij})^2$

4. Compute leading eigenvectors $w_1 \ldots$ and eigenvals $\lambda_1 \ldots$ of C $\quad Z^T = (w^1 \ldots w^k) \begin{pmatrix} \sqrt{\lambda_1} & & \\ & \ddots & \\ & & \sqrt{\lambda_k} \end{pmatrix}$

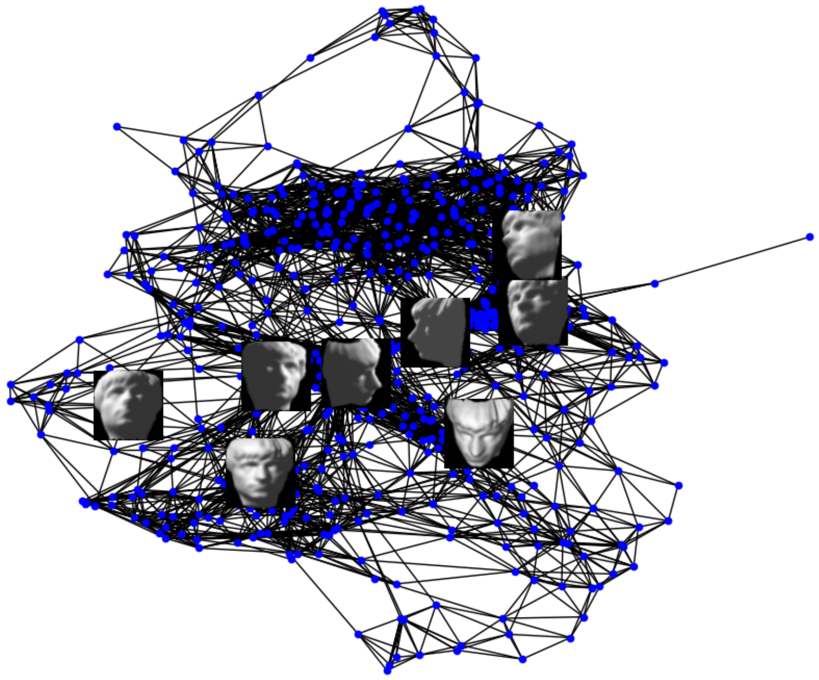
$C = U \Lambda U^T \to$ eigendecomp

1. The adjacency matrix plot and the nearest neighbor graphs represent the faces represented visually relative to each other.

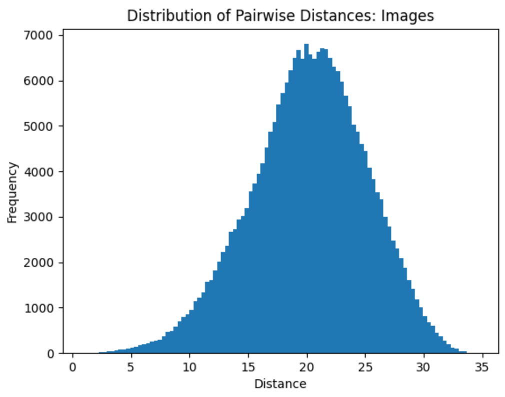
   The data matrix is transposed to maintain the appropriate format of M x K where M is number of images and each column represents a pixel. K being 4096 is a flattened representation of the 64 x 64 sized images.

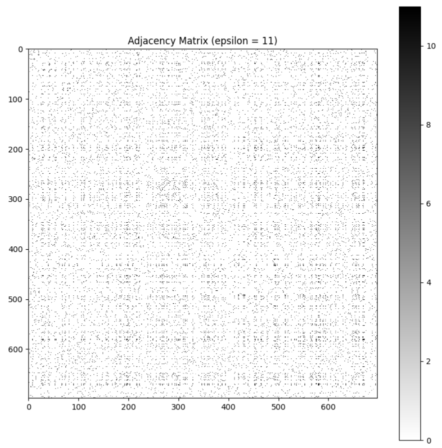
   The distance matrix (used to visualize the graph as well as compute the adjacency matrix) is computed using a Scipy function with the l2norm Euclidean distance used as the distance metric. The distances are then plotted on the graph below along with the corresponding faces for a few of them:

Graph with Random Images Plotted Near Nodes

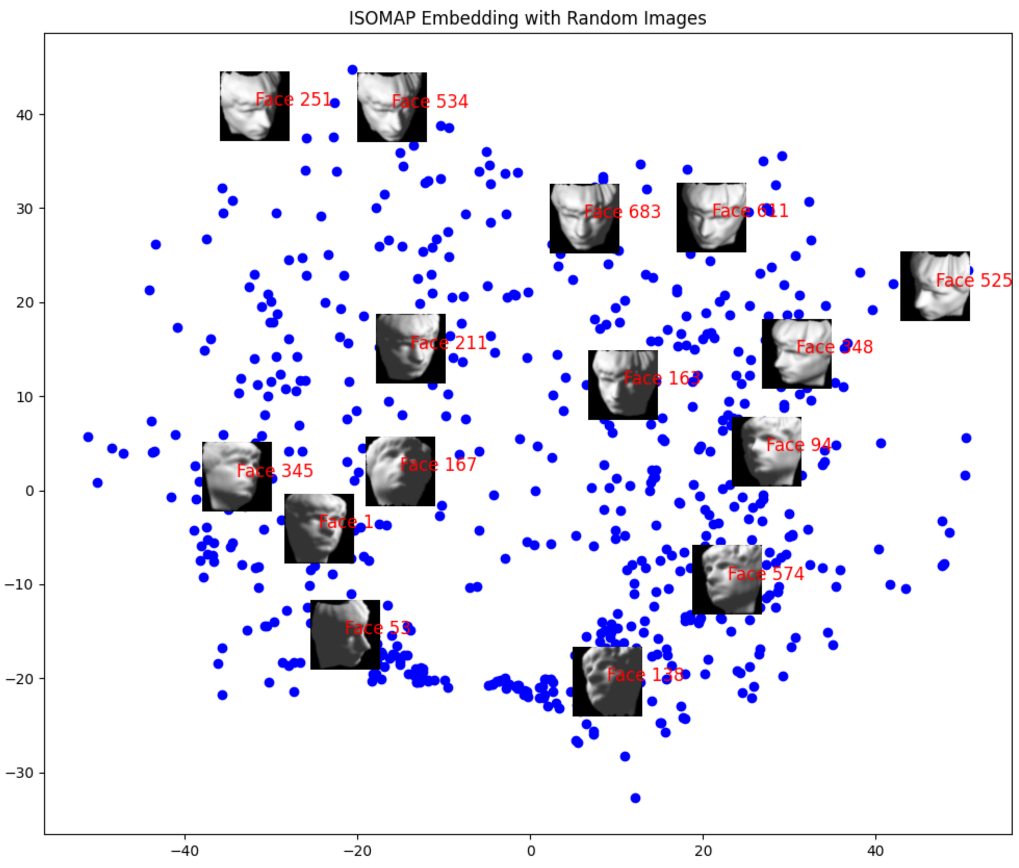


The adjacency matrix below corresponds to the adjacency matrix with the lowest epsilon (Epsilon = 11) where we have one connected component. To arrive at this figure, we computed the adjacency matrix for every value of Epsilon corresponding to the Euclidean distance between each pair, from the min of 1 to a max of 35.



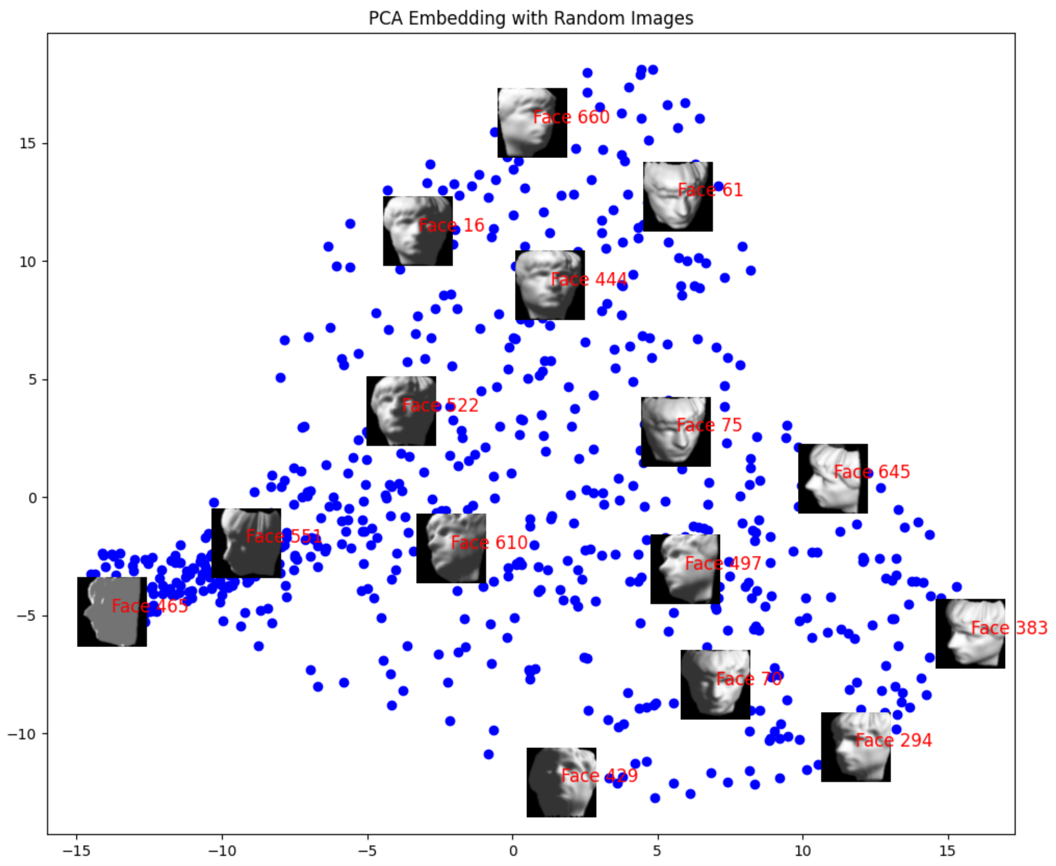


2. The next step in implementing our ISOMAP algorithm was to compute the pairwise shortest distance matrix D which holds the shortest distance between any two observations from all the pair combination distances represented in the adjacency matrix. To compute the geodesci distances, the Djikstra shortest distance algorithm from Scipy's shortest_path function was used. Given the distance between observation a and b was equivalent to the distance between observation b and a, our graph was treated as an un-directed network. The next step was to implement the MDS method as shown in the course lectures to calculate the centered matrix H and then the covariance matrix C used for eigen-decomposition. Once the eigenvalues and eigenvectors were computed, the final step before visualization was to get the two dimensional embedding by multiplying the top two eigenvectors with the square root of the two eigenvalues respectively.



In the figure above showing the top components (embedding) visualized in a two dimension plot with corresponding faces added in, it seems that the faces on the left favor facing to the right whereas as those on the right seem to face to the left. Additionally, it does seem the faces on the bottom are darker or have more shadows whereas the images on the top have higher levels of contrast. It also seems that images higher seem to be looking down whereas lower images favor looking straight out or even up. Although the axes may be reversed, it does seem to track with the findings of the ISOMAP paper (Course link in Sources section).

3. The sklearn implementation of PCA was used to generate two dimension projections to compare performance with ISOMAP. the visualization of the embedding with corresponding images for a select number of projected data points is displayed below. The PCA implementation seems to separate out darker images with lower contrast to the bottom left with higher contrast and brighter images to the bottom right. It does seem more difficult to interpret face direction by quadrant or other meaningful demarcation than the ISOMAP rendering. As expected, ISOMAP seems to provide a more visually meaningful projection.


PCA Embedding with Random Images
Face 660
Face 61
Face 16
Face 444
Face 522
Face 75
Face 645
Face 551
Face 610
Face 497
Face 465
Face 383
Face 70
Face 294
Face 439
15
10
5
0
−5
−10
−15
−10
−5
0
5
10
15

4 Eigenfaces and simple face recognition

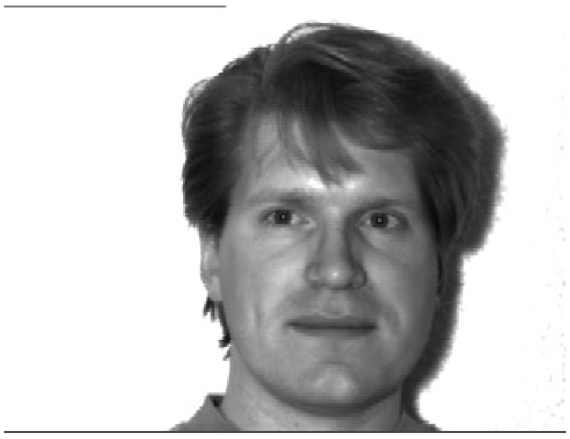


Figure 1: Subject 1

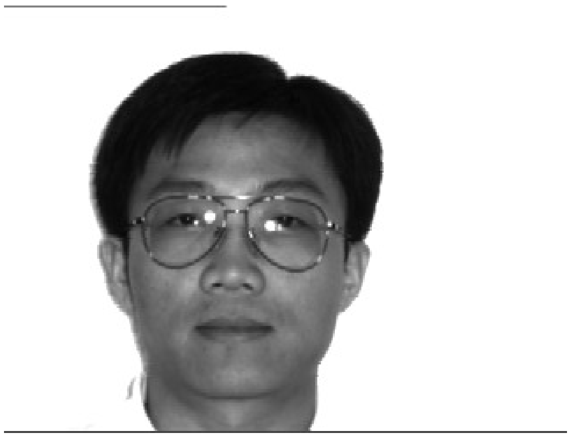
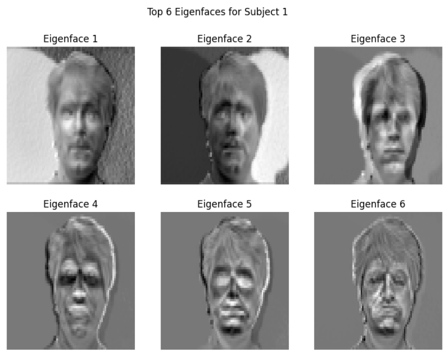
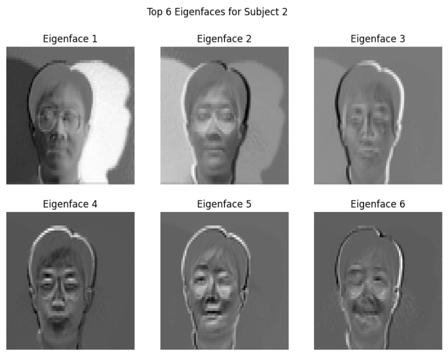


Figure 2: Subject 2

1. For this section, we performed PCA on the Yale faces data set to visualize the top 6 eigenfaces for the test faces provided and then calculate the projection residual between the test faces and the top eigenface to perform a facial recognition task. To obtain the top 6 eigenfaces, the two groups of images, once imported, properly flattened, and converted into arrays, were then centered before having PCA performed on them using the standard Scikit-learn library implementation.The first 6 components corresponding to the training data for the test image were obtained and then reshaped back into image format to be displayed below.



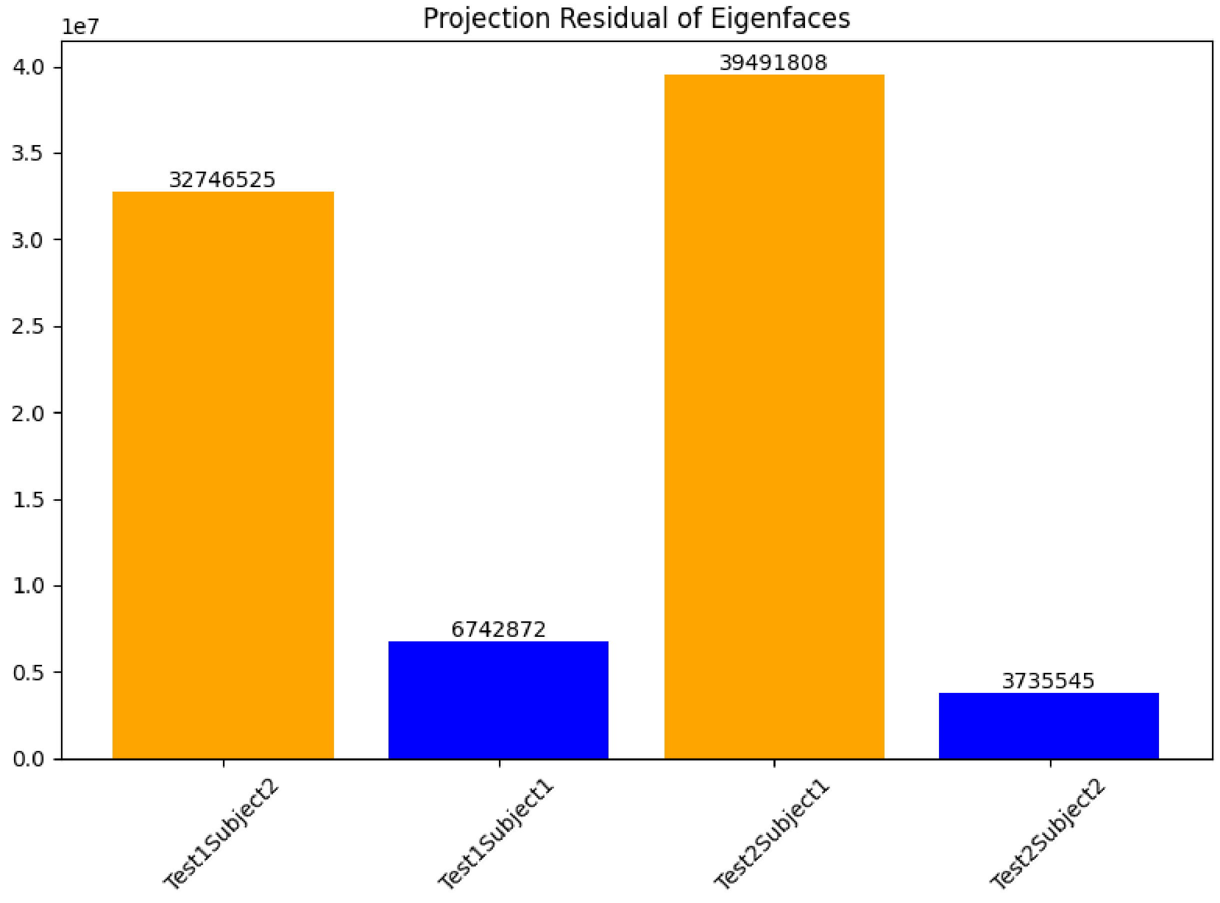


The top 6 eigenfaces do have visual structure and appear to be like faces. They correspond to the most informative components relative to largest variance in descending order. This can be seen visually as the first component for both subjects appears to be the most visually distinct and faithful to original representations, the second less so and the last seems to be the most homogenous component for each subject, appearing the least like a face and with the least difference between facial components and the background.

2. We then used the top eigenface as well as the mean image vector we calculated earlier for each subject to calculate the projection residual via l2norm-squared distance as shown by the equation below:

$$s_{ij} = \left\| (\text{test image})_j - (\text{eigenface}_i)(\text{eigenface}_i)^T (\text{test image})_j \right\|_2^2 \tag{1}$$

The plot below shows the projection residuals of the test images j performed with the mean image vectors and top eigenfaces for Subject i. Where j == i (shown in blue), the projection residuals are far lower than for the situations where j != i (shown in orange). That's because in those cases the test image belongs to the same Subject as the mean image vector and top eigenface. You can then use the top eigenface to determine which set of images the test image belongs to.



3. The algorithm seemed to perform well on the data we provided given that we can clearly tell which Subject each test image should belong to by analyzing the projection residual score. As in the previous problem, PCA may not be the most effective dimensionality reduction technique for data where linear distances aren't appropriate. It may be worth exploring other methods of generating projected residuals, such as using ISOMAP or another algorithm to generate the top eigenfaces and mean image vectors used in our calculation.

Sources

1. https://gatech.instructure.com/courses/421726/files/52795125?module_item_id=42

2. https://gatech.instructure.com/courses/421726/files/52795125?module_item_id=4240714

3. Ed Discussions

4. NetworkX documentation

5. Sci-py documentation

6. Numpy documentation

7. Matplotlib documentation

8. Pillow documentation

9. https://www.youtube.com/watch?v=UyAfmAZU_WI

10. https://jonathan-hui.medium.com/machine-learning-singular-value-decomposition-svd-principal-component-analysis-pca-1d45e885e491

11. https://stats.stackexchange.com/questions/134282/relationship-between-svd-and-pca-how-to-use-svd-to-perform-pca

12. Course Lectures and Demo Code

13. https://www.geo.fu-berlin.de/en/v/soga-py/Advanced-statistics/Multivariate-Approaches/ Principal-Component-Analysis/PCA-the-basics/Choose-Principal-Components/index.html#:~:text=A%20widely %20applied%20approach%20is,elbow%20in%20the%20scree%20plot.

14. Simulation ISYE6644 Notes (MLE)