ISYE 6740 Summer 2021

Homework 2

(100 points + 12 bonus points)

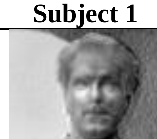
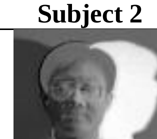
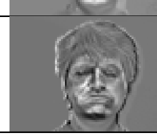
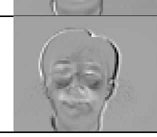
1. Eigenfaces and simple face recognition [52 points; including 2 bonus points.]

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.

(a) (25 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?

No.	Subject 1	Subject 2
1. Eigen face		
2. Eigen face		
3. Eigen face		
4. Eigen face		
5. Eigen face		
6. Eigen face		

The above eigne faces capture different aspects of all the pictures from each subject:

- Subject 1
  - Eigenface 1 captured most deviation from the mean in terms of overall face features and structure
  - Eigenface 2 captured lighting on the face better than first eigenface
  - Other Eigenfaces faces captured combinations of multiple face expressions and facial feature outlines

As expected, the quality of features is higher/better in the order of the eigen values or eigen faces. I.e, Eigen face1 capture more information than eigen face 6

(b) (25 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.

<u>**Projection residual**</u>	**Test image 1**	**Test image 2**
Eigen Face 1	S11 = 1,692,209	S12 = 8,401,524
Eigen face 2	S21 = 18,505,248	S22 = 4,454,498
	Belongs to subject 1	Belongs to subject 2

The lower the projection residual, the closer the test image is to the eigen face, since we are calculating the distance between them. This is because Lower the projection residual value, implies that the eigen vector explains the features of the test vector better. Hence the test image is closer to the eigen face.
This implies that a test image belongs to the subject which has lowest projection residual on the subjects' top eigen face.

Based on this,

- S11 < S21 implies, Test image 1 belongs to Subject 1
- S22 < S12 implies, Test image 2 belongs to Subject 2

(c) (Bonus: 2 points) Explain if face recognition can work well and discuss how we can improve it possibly.

The face recognition works alright for the clear front views, but if we build the eigen faces from more high definition images and include more variations in the training data (i.e, with side slightly side poses, flipped images, distorted, different scales etc), we can achieve better Face ID results.

2. Order of faces using ISOMAP [50 points]

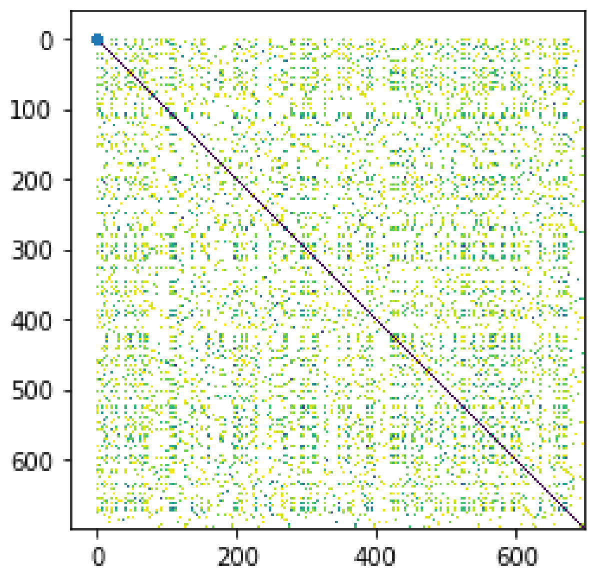
This question aims to reproduce the ISOMAP algorithm results in the original paper for ISOMAP, J.B. Tenenbaum, V. de Silva, and J.C. Langford, Science 290 (2000) 2319-2323 that we have also seen in the lecture as an exercise (isn't this exciting to go through the process of generating results for a high-impact 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 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 R^{4096}) to construct the E-ISOMAP (follow the instructions in the slides.) You will tune the E 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).

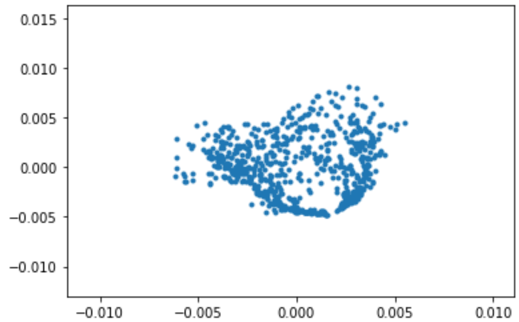
The nearest neighbor graph as a Weighted Adjacency matrix is picturized below. The axes are - 698 images as they are in the isomap.mat data along both X and Y axes.



The darker the color, higher the distance. Blank data (zero values) imply that the pair of images are not in the close (Epsilon) distance range.

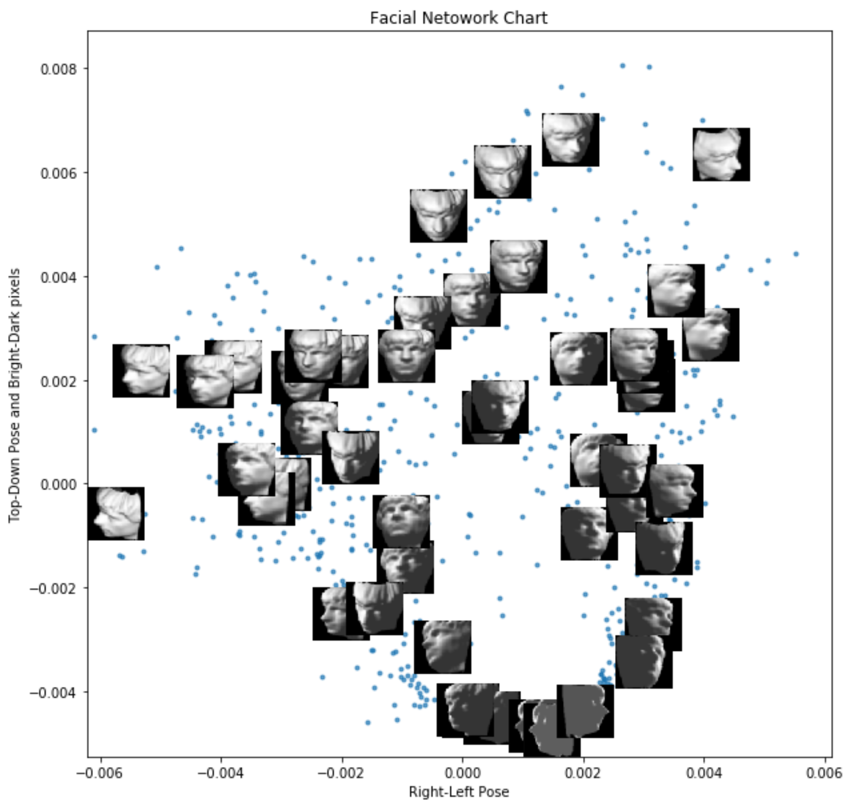
Shortest Distance metric = 'Euclidean'
The distance used for weighing distance between image pairs = Epsilon (Tuned) = 15 units

(b) (20 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?



The above is the scatter plot of the Two dimensional (low dim) embedding (Z) of the ISOMAP.
Picked 50 random data points and plotted their actual images from the dataset below.

D = Euclidean, Epsilon = 15

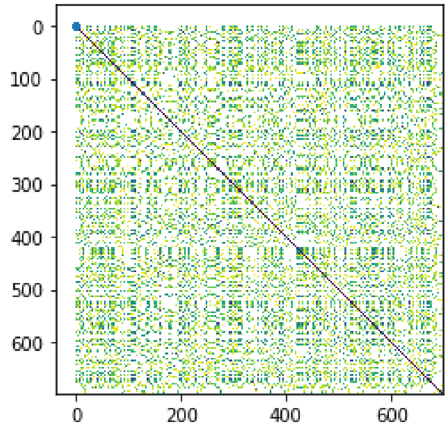


As mentioned on the axes we notice that;

- Y axis:
  - Top to bottom the face rotate from a top view to bottom view
  - Top to bottom the images lighting changes from lighter to darker
- X-axis:
  - Left to right the face rotates from left to right views

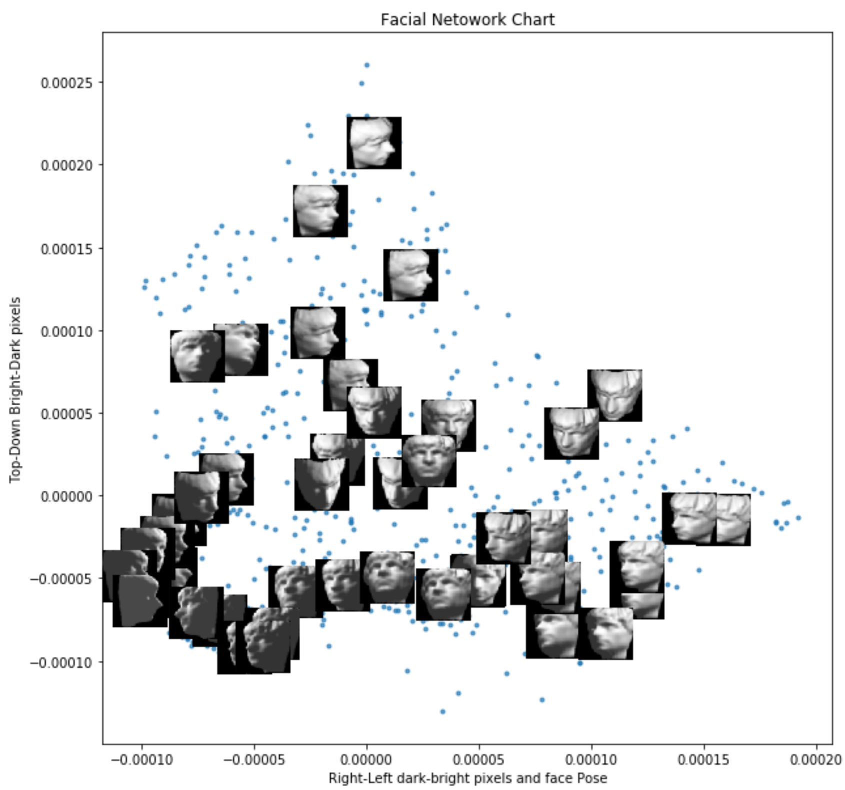
(c) (10 points) Now choose ℓ_1 distance (or Manhattan distance) between images (recall the definition from "Clustering" lecture)). Repeat the steps above. Use ϵ-ISOMAP to obtain a $k = 2$ dimensional embedding. Present a plot of this embedding and specify the face locations on the scatter plot. Do you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

Repeating the steps above, The nearest neighbor graph as a Weighted Adjacency matrix is picturized below.
Shortest Distance metric = 'Manhattan'
The distance used for weighing distance between image pairs = Epsilon (Tuned) = 800 units



Below is the scatter plot of the Two dimensional embedding (Z) of the ISOMAP by picking 50 random data points and plotted their actual images from the dataset below

D = Manhattan, Epsilon = 800

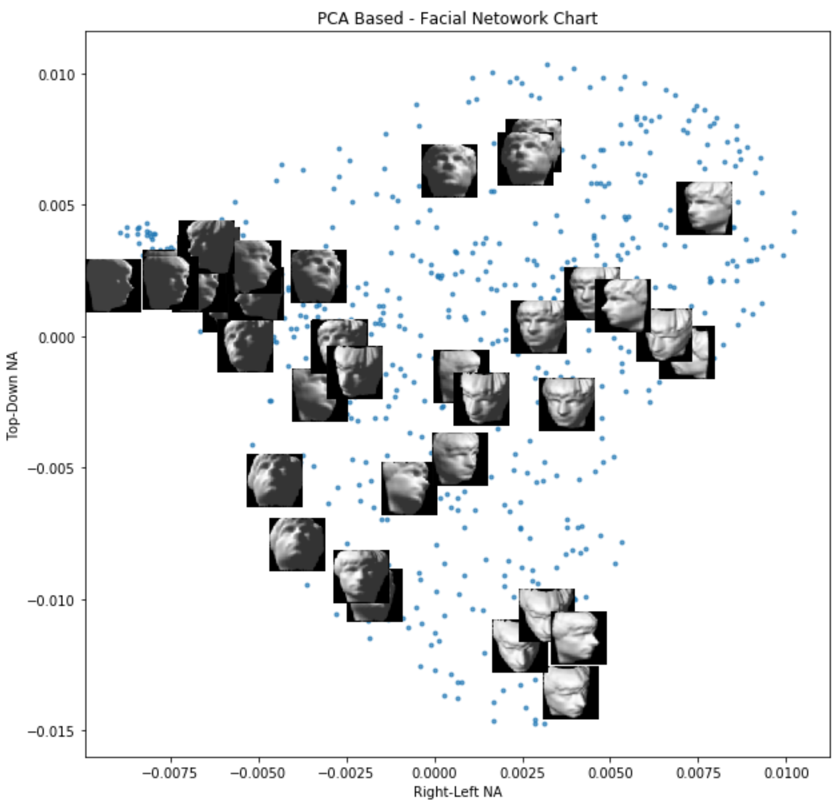


Using Manhattan (City block) distance does a good job in terms of capturing the geometry between the pictures. In contrast with the Euclidean distance, the principal axes captures the variances a little differently as;

- Y axis:
  - Top to bottom the pixels change from lighter to darker shades.
- X-axis:
  - Left to right the image lighting changes from lighter to darker, and also
  - The face rotates from left to right views

This categorization is not as accurate as with the Euclidean measure. If we notice some faces to the bottom left, they are facing opposite to what the X-axis trend is.

(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.



The above is the scatter plot of the principal components determined from PCA and the images projected on to them. From this Facial network chart, it seems like the X-axis (First component) is capturing the lighting conditions of dark-brighter but the second component's feature is not very interpretable. None of these two components are capturing the face pose / rotation.

ISO MAP does a better job in terms of capturing the variances in the images. This implies the data images have some kind of non-linear combination of features. So, the manifold learning did a better job than linear dimensionality reduction.

3. PCA: Food consumption in European countries [Bonus Question: 10 points]

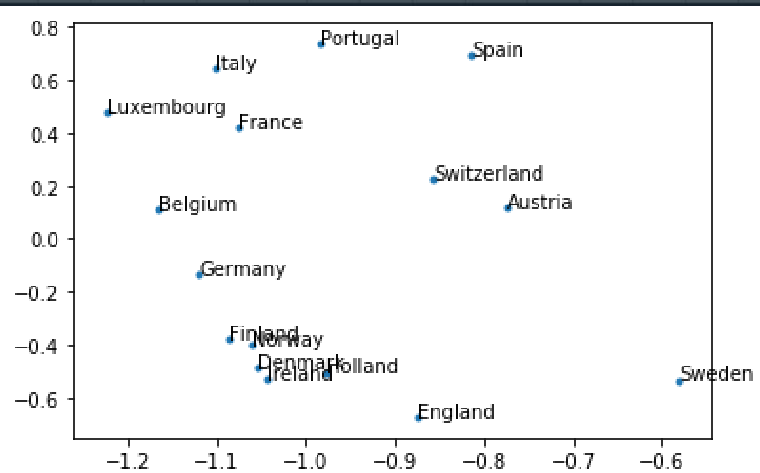
The data food-consumption.csv contains 16 countries in Europe and their consumption for 20 food items, such as tea, jam, coffee, yogurt, and others. We will perform principal component analysis to explore the data. In this question, please implement PCA by writing your own code (you can use any basic 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) (5 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.

Row of Matrix = Each country's food consumption feature
Column of Matrix = Country

Index	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Country	Germany	Italy	France	Holland	Belgium	Luxembourg	England	Portugal	Austria	Switzerland	Sweden	Denmark	Norway	Finland	Spain	Ireland
Real coffee	90	82	88	96	94	97	27	72	55	73	97	96	92	98	70	30
Instant coffee	49	10	42	62	38	61	86	26	31	72	13	17	17	12	40	52
Tea	88	60	63	98	48	86	99	77	61	85	93	92	83	84	40	99
Sweetener	19	2	4	32	11	28	22	2	15	25	31	35	13	20	18	11
Biscuits	57	55	76	62	74	79	91	22	29	31	61	66	62	64	62	80
Powder soup	51	41	53	67	37	73	55	34	33	69	43	32	51	27	43	75
Tin soup	19	3	11	43	23	12	76	1	1	10	43	17	4	10	2	18
Potatoes	21	2	23	7	9	7	17	5	5	17	39	11	17	8	14	2
Frozen fish	27	4	11	14	13	26	20	20	15	19	54	51	30	18	23	5
Frozen veggies	21	2	5	14	12	23	24	3	11	15	45	42	15	12	7	3
Apples	81	67	87	83	76	85	76	22	49	79	56	81	61	50	59	57
Oranges	75	71	84	89	76	94	68	51	42	70	78	72	72	57	77	52
Tinned fruit	44	9	40	61	42	83	89	8	14	46	53	50	34	22	30	46
Jam	71	46	45	81	57	20	91	16	41	61	75	64	51	37	38	89
Garlic	22	80	88	15	29	91	11	89	51	64	9	11	11	15	86	5
Butter	91	66	94	31	84	94	95	65	51	82	68	92	63	96	44	97
Margarine	85	24	47	97	80	94	94	78	72	48	32	91	94	94	51	25
Olive oil	74	94	36	13	83	84	57	92	28	61	48	30	28	17	91	31
Yoghurt	30	5	57	53	20	31	11	6	13	48	2	11	2	21	16	3
Crisp bread	26	18	3	15	5	24	28	9	11	30	93	34	62	64	13	9

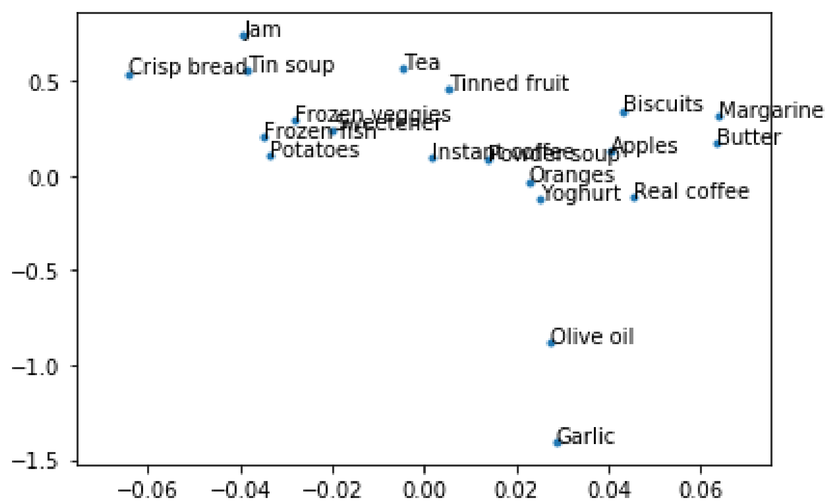


From the above graph we can infer that the Norwegian nations (Finland, Norway, Denmark, Holland and Ireland) have very similar food consumption pattern. But Sweden separates itself not just from the geographically close Norwegian nations but all the other nations, by a rather unique food consumption style. Switzerland and Australia again seem to have similar styles but different from other nations. Luxembourg and Spain have unique patterns too!
Quantitatively, England and Norwegian countries (-0.4 to -0.6 PC2) have sort of opposite consumption patterns with France, Italy, Spain, Portugal and Luxembourg (+0.4 to +0.8 PC2).

(b) (5 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.

Index	Country	Real coffee	Instant coffee	Tea	Sweetener	Biscuits	Powder soup	Tin soup	Potatoes	Frozen fish	Frozen veggies	Apples	Oranges	Tinned fruit	Jam	Garlic	Butter
0	Germany	90	49	88	19	57	51	19	21	27	21	81	75	44	71	22	91
1	Italy	82	10	60	2	55	41	3	2	4	2	67	71	9	46	80	66
2	France	88	42	63	4	76	53	11	23	11	5	87	84	40	45	88	94
3	Holland	96	62	98	32	62	67	43	7	14	14	83	89	61	81	15	31
4	Belgium	94	38	48	11	74	37	23	9	13	12	76	76	42	57	29	84
5	Luxembourg	97	61	86	28	79	73	12	7	26	23	85	94	83	20	91	94
6	England	27	86	99	22	91	55	76	17	20	24	76	68	89	91	11	95
7	Portugal	72	26	77	2	22	34	1	5	20	3	22	51	8	16	89	65
8	Austria	55	31	61	15	29	33	1	5	15	11	49	42	14	41	51	51
9	Switzerland	73	72	85	25	31	69	10	17	19	15	79	70	46	61	64	82
10	Sweden	97	13	93	31	61	43	43	39	54	45	56	78	53	75	9	68
11	Denmark	96	17	92	35	66	32	17	11	51	42	81	72	50	64	11	92
12	Norway	92	17	83	13	62	51	4	17	30	15	61	72	34	51	11	63
13	Finland	98	12	84	20	64	27	10	8	18	12	50	57	22	37	15	96
14	Spain	70	40	40	18	62	43	2	14	23	7	59	77	30	38	86	44
15	Ireland	30	52	99	11	80	75	18	2	5	3	57	52	46	89	5	97

Row of Matrix = Country
Column of Matrix = Each country's food consumption feature



Considering all the 16 Countries food consumption patterns, we notice that Garlic stands out to be an eccentric food item. It's consumption pattern across countries is unique. However, Margarine-Butter are very closely related and as expected their consumption is similar Similarly all the frozen food and potatoes which has higher shelf life are lumped together indicating most countries having one of them has either of them in their diet.

All the closely grouped food items are consumed in a cimilar fashion across the countries and the ones that are away could be the eccentric choices by a few countries .