Name: Jianyuan Lu GT ID: 903633385

1.PCA: Food consumption

1.1

Now the data matrix $\mathbf{X}$ become like this: the shape of $\mathbf{X}$ is (20,16), which means the total row number is 16, and the total column number is 20. The rows represent the features of the each country, such as the Real coffee and Instant coffee. The columns represent the samples, specifically, the countries that I am studying for, such as Germany and Italy.

1.2

Suppose $\mathbf{X} = (X_1, X_2, \ldots . X_n)^T$, which is an n dimensional random variable vector. Since the PCA is like the linear transformation, which transform $\mathbf{X}$ to $\mathbf{Z}$=$(Z_1, Z_2, \ldots . Z_n)^T$ and is shown as below:

$$\begin{cases} Z_1 = \mathbf{w}_1^T\mathbf{X} = w_{11}X_1 + w_{12}X_2 + \cdots + w_{1n}X_n \\ Z_2 = \mathbf{w}_2^T\mathbf{X} = w_{21}X_1 + w_{22}X_2 + \cdots + w_{2n}X_n \\ \vdots \\ Z_n = \mathbf{w}_n^T\mathbf{X} = w_{n1}X_1 + w_{n2}X_2 + \cdots + w_{nn}X_n \end{cases}$$

Based on the principle that we want to let Z_i to store the max information of $\mathbf{X}$, which means $\mathrm{Var}(Z_i)$ become max.

$\mathrm{Var}(Z_i) = \frac{1}{n}\mathbf{w}_1^T(\mathbf{X} - \overline{\mathbf{X}})(\mathbf{X} - \overline{\mathbf{X}})^T\mathbf{w}_1$, where $\overline{\mathbf{X}}$ is the mean vector of $\mathbf{X}$.

Let $\mathbf{C} = \frac{1}{n}(\mathbf{X} - \overline{\mathbf{X}})(\mathbf{X} - \overline{\mathbf{X}})^T$,which is the covariance matrix of $\mathbf{X}$. So the $\mathrm{Var}(Z_1) = \boldsymbol{w_1^T}\mathbf{C}\mathbf{w}_1$, and the problem is find the $\mathbf{w}_1$ which can maximize $\mathrm{Var}(Z_1)$, where letting $\|\boldsymbol{w}_1\|^2 \leq 1$. This is a conditional optimal problem, so we can use the Lagrangian optimization:

$$L(\mathbf{w}_1, \lambda) = \boldsymbol{w_1^T}\mathbf{C}\mathbf{w}_1 + \lambda(1 - \|\boldsymbol{w}_1\|^2)$$

$$\begin{cases} \dfrac{\partial L}{\partial \mathbf{w}_1} = 2\mathbf{C}\mathbf{w}_1 - 2\lambda\mathbf{w}_1 = 0 \\ \dfrac{\partial L}{\partial \lambda} = 1 - \|\boldsymbol{w}_1\|^2 = 0 \end{cases}$$

From the upper equation, we know that $\|\boldsymbol{w}_1\|^2 = 1$ and $\mathbf{w}_1$ is the eigenvalue of $\mathbf{C}$ and λ is

the corresponding eigenvalue.

So $\mathrm{Var}(Z_1) = \boldsymbol{w_1^T \mathbf{C} \mathbf{w}_1} = \lambda \|\boldsymbol{w}_1\|^2 = \lambda$, and the problem becomes to find the largest eigenvalue and corresponding eigenvector.

Then, if we use the eigendecomposition of $\mathbf{C}$ and find the largest eigenvalue and corresponding eigenvector, the next step is to find the rest of the principle component, such as Z_2. Since we don't want Z_2 to show the same information which Z_1 has, so Z_2 and Z_1 are orthogonal, which means $Cov(Z_2, Z_1) = \boldsymbol{w_2^T \mathbf{C} \mathbf{w}_1} = 0$. So when we try to maximize $\mathrm{Var}(Z_2)$ with respect to $\mathbf{w}_2$, we have a extra condition, which is $\boldsymbol{w_2^T \mathbf{C} \mathbf{w}_1} = 0$. However, since each eigenvector is orthogonal with each other, this condition can be automatically satisfied. So we can just select the i-th eigenvector as $\mathbf{w}_i$ and compute the i-th principle component $Z_i = \mathbf{w}_i^T \mathbf{X}$, where the corresponding eigenvalue is the i-th largest.

1.3

The first two principal two component for each data point(country) is:

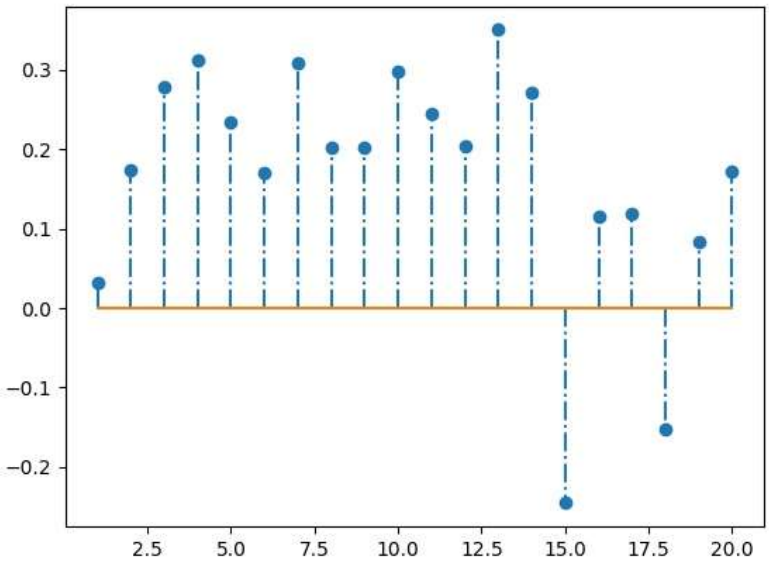
country	First component $Z1$	Second component $Z2$
Germany	0.546	0.060
Italy	-1.497	0.221
France	-0.246	1.183
Holland	1.094	0.907
Belgium	-0.307	0.429
Luxembourg	0.665	0.999
England	1.604	0.947
Portugal	-1.910	-0.522
Austria	-1.279	-0.455
Switzerland	0.201	0.795

Sweden	1.335	-2.224
Denmark	0.988	-1.359
Norway	0.040	-1.064
Finland	-0.289	-1.196
Spain	-0.961	0.214
Ireland	0.016	1.065

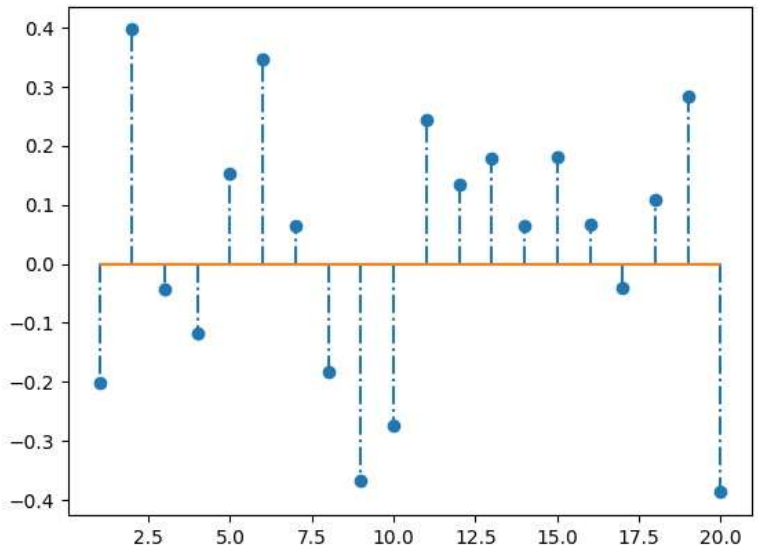
The weighted vector w_1 and w_2 are shown as below:

weighted vector	w_1	w_2
eigenvalue	51107	31277
Real coffee	0.032	-0.201
Instant coffee	0.173	0.397
Tea	0.279	-0.044
Sweetener	0.312	-0.118
Biscuits	0.233	0.153
Powder soup	0.171	0.346
Tin soup	0.309	0.064
Potatoes	0.202	-0.184
Frozen fish	0.202	-0.368
Frozen veggies	0.298	-0.274
Apples	0.244	0.245
Oranges	0.204	0.134
Tinned fruit	0.350	0.178
Jam	0.272	0.064
Garlic	-0.245	0.181
Butter	0.115	0.066
Margarine	0.118	-0.042
Olive oil	-0.152	0.109
Yoghurt	0.083	0.283
Crisp bread	0.171	-0.387

The graph for w_1:



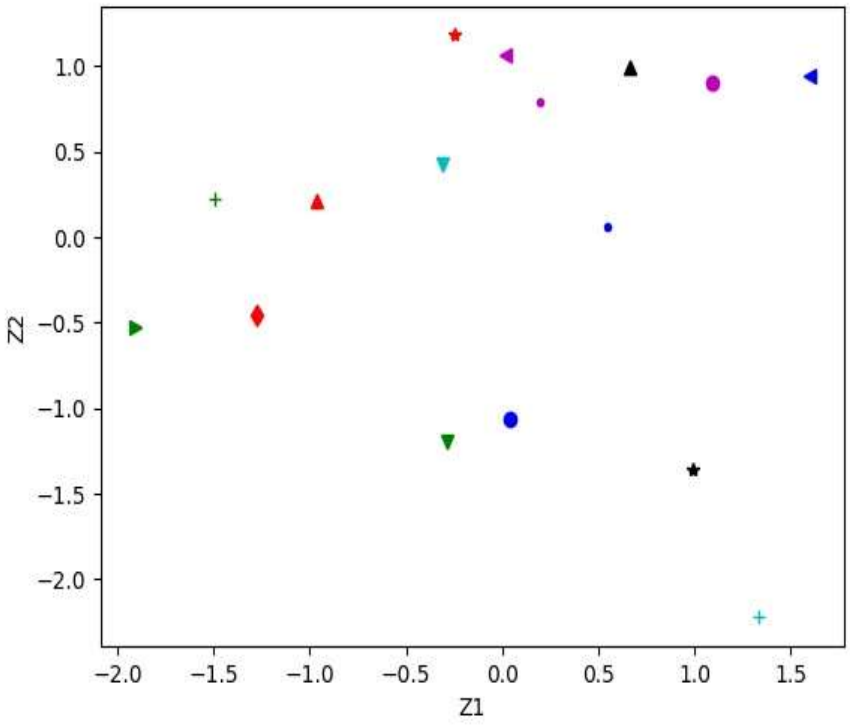
The graph for w_2:



From the graph of w_1, which has the largest eigenvalue, since almost every component is positive, we could know that if the country consumes more Real coffee, this country will tend to consume more other types of food, such as Instant coffee, Tea, Powder soup and so on(except for Garlic and Olive oil). So we could call the first component the “consume index”, which indict that if the “consume index” is higher, this country tend to consume more food no matter the type of food(except for Garlic and Olive oil).

From the graph of w_2, which has the second largest eigenvalue, indicate that if w_2 is higher, this country tends to consume more food like Instant coffee, Powder soup, Yoghurt and some fruit and fresh vegetables, and tend to consume less unhealthy food such as Potatoes, Frozen fish, Frozen veggies, Margarine and Crisp bread. So, we could call the second component the "healthy food index".

1.4



Country	Legend
Germany	●
Italy	+
France	★
Holland	●
Belgium	▼
Luxembourg	▲
England	◀
Portugal	▶
Austria	♦
Switzerland	●
Sweden	+
Denmark	★
Norway	●
Finland	▼
Spain	▲
Ireland	◀

As I referred from 1.3, $Z1$ is the consume index which indicate the total amount of the consumption food. So England, Luxembourg, Holland, Denmark consume less garlic and olive oil and consume more other type of food. On the contrast, Portugal, Italy and Austria consume more garlic and olive oil and consume less other type of food.

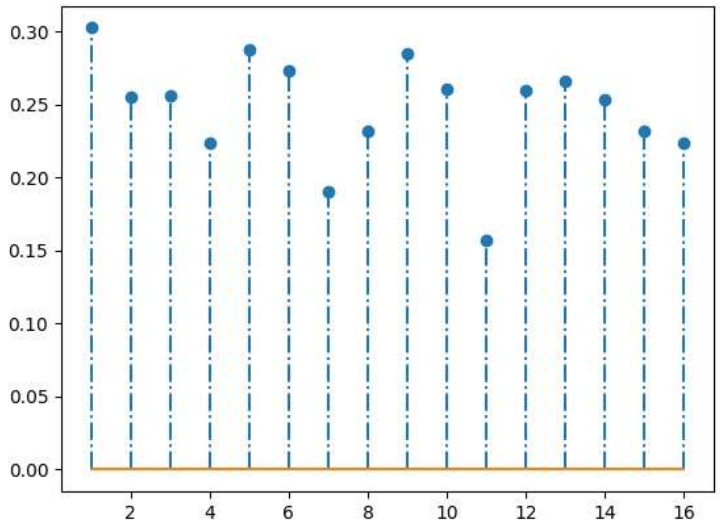
$Z2$ is the healthy food index which indicate the proportion of healthy food in the total amount of food. So England, Luxembourg, Holland, France and Ireland eat more fresh food and vegetables while Sweden, Denmark, Norway and Finland eat more frozen and tin food, maybe this is related to the weather and location.

1.5

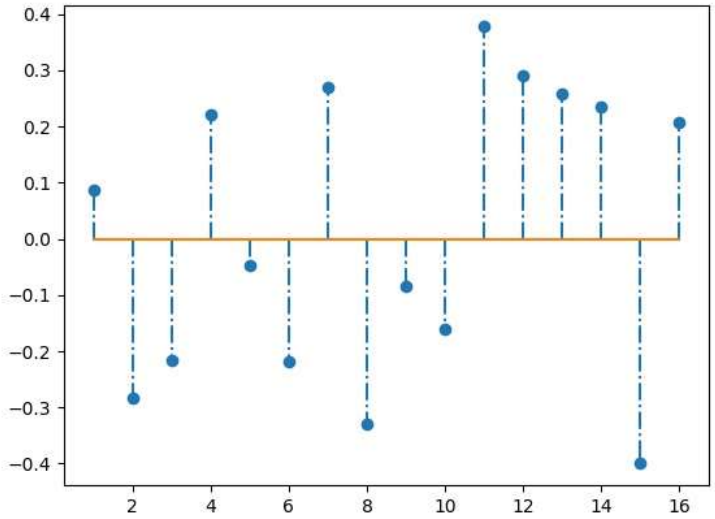
The first two principal two component for each data point(food) is:

Food	First component $Z1$	Second component $Z2$
Real coffee	1.494	-0.008
Instant coffee	-0.279	-0.463
Tea	1.392	1.059
Sweetener	-1.217	0.171
Biscuits	0.584	0.455
Powder soup	0.132	-0.217
Tin soup	-1.304	0.744
Potatoes	-1.456	-0.039
Frozen fish	-1.053	0.294
Frozen veggies	-1.322	0.399
Apples	0.963	0.004
Oranges	1.079	-0.176
Tinned fruit	-0.237	0.575
Jam	0.373	1.138
Garlic	-0.062	-3.167
Butter	1.325	0.248
Margarine	1.093	0.470
Olive oil	0.399	-1.946
Yoghurt	-1.061	-0.698
Crisp bread	-0.842	1.157

The graph for w_1:

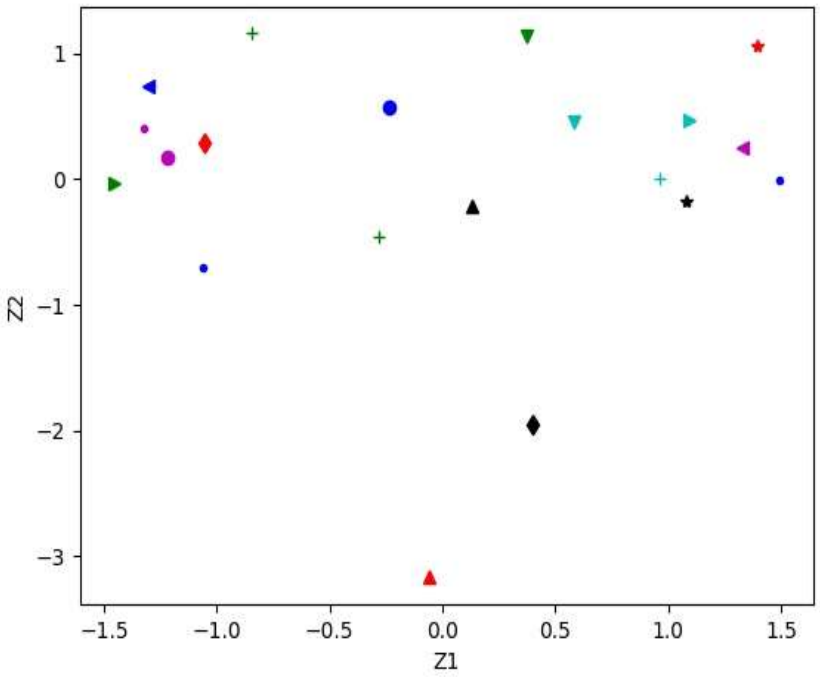


The graph for w_2:



From the graph for w_1, we know that all the countries share the same amount of consumption for one particular food. That is to say, for a specific type of food, if one country consume a large amount of this food, the other countries tend to consume a large amount as well.

The graph for w_2 shows the variance between countries when considering the specific type of food consumption.



Food	Legend
Real coffee	
Instant coffee	
Tea	
Sweetener	
Biscuits	
Powder soup	
Tin soup	
Potatoes	
Frozen fish	
Frozen veggies	
Apples	
Oranges	
Tinned fruit	
Jam	
Garlic	
Butter	
Margarine	
Olive oil	
Yoghurt	
Crisp bread	

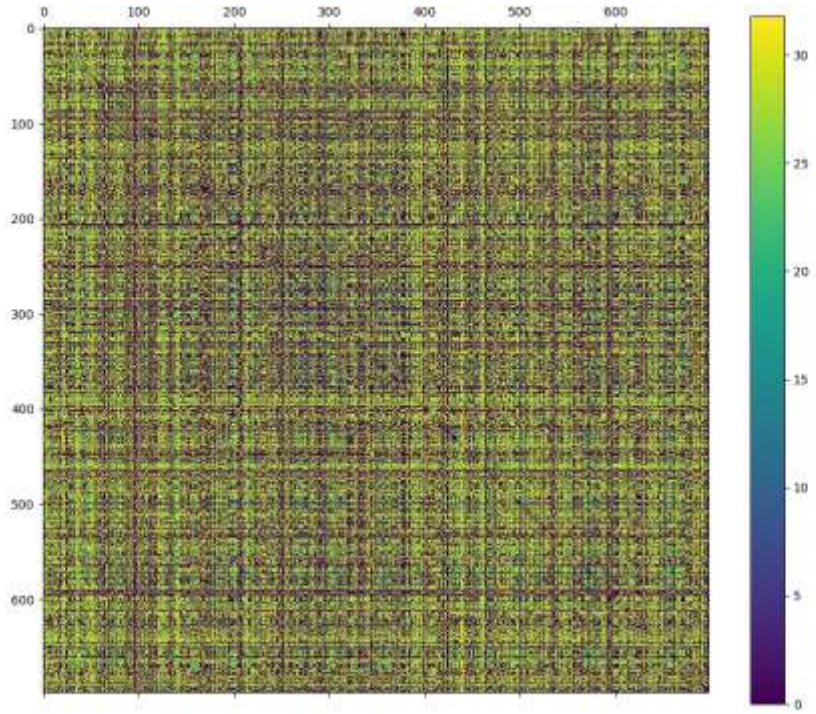
From the upper scatter graph and table, we know that the consumption of Tea, Butter, Real coffee and Oranges is larger in all the countries when considering the $Z1$ scale. When considering the $Z2$ scale, we also know that for the food such as Instant coffee, Jam and Tea, Italy, Portugal and Spain don't like to them.

2 Order of faces

(a)

I take two strategies to form my similarity graph:

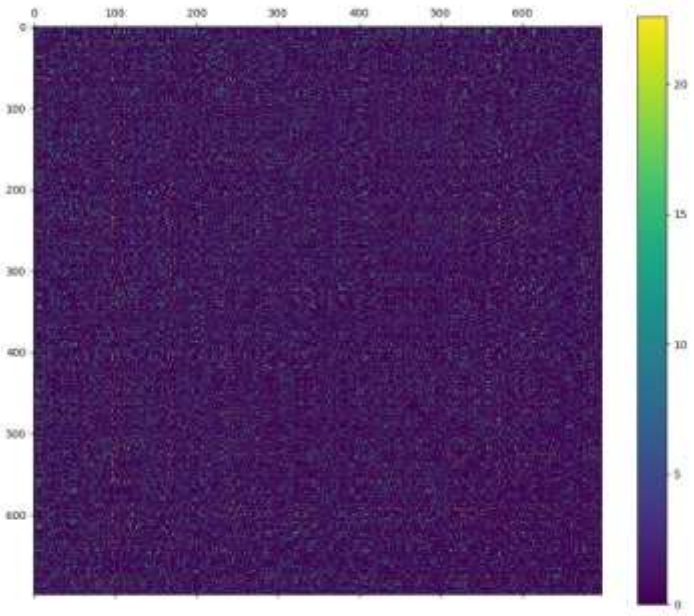
1. The first one is to set the global ϵ to find the neighbors for all nodes, which guarantees that each node should have at least 100 neighbors. Since this is the global ϵ, so the value of ϵ is very high and many nodes have 400-500 neighbors. The similarity matrix is shown like this:



As you can see, the lighter the graph is, the higher the distance is, and many nodes have much more neighbors(about 400-500), which highly exceeds the threshold(100). Since there are so many nodes which have many neighbors, so the strategy one will still break the original data space and have a similar look with PCA analysis.

2. The second strategy is to set the local ϵ for each node, which just guarantees that each

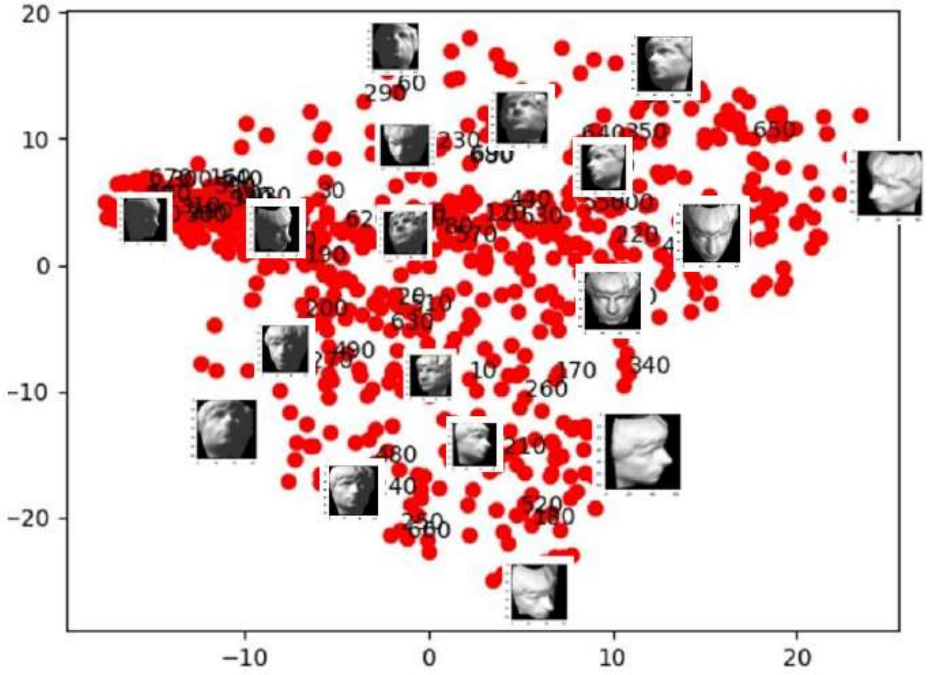
node has at least 100 neighbors, and each node's threshold ϵ is different depending on whether this node is in a dense area or not. The similarity matrix is shown like this:



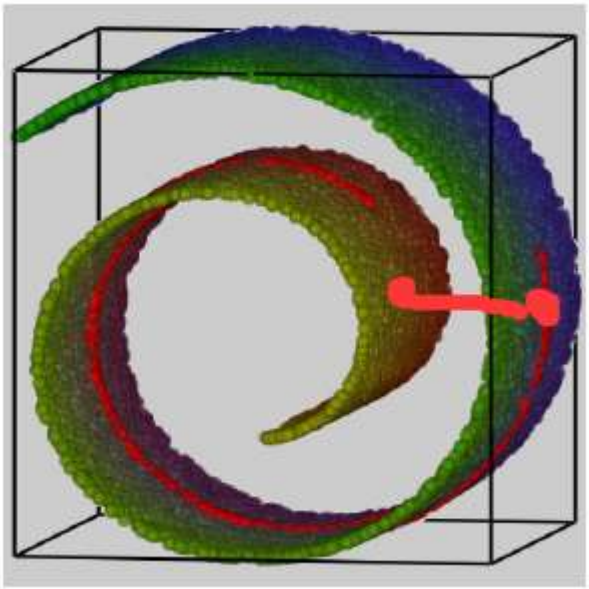
As you can see, many parts of the matrix is 0, which indicate that those two points are not connected. The second strategy preserves the original data space and guarantees that each node has at least 100 neighbors.

(b)

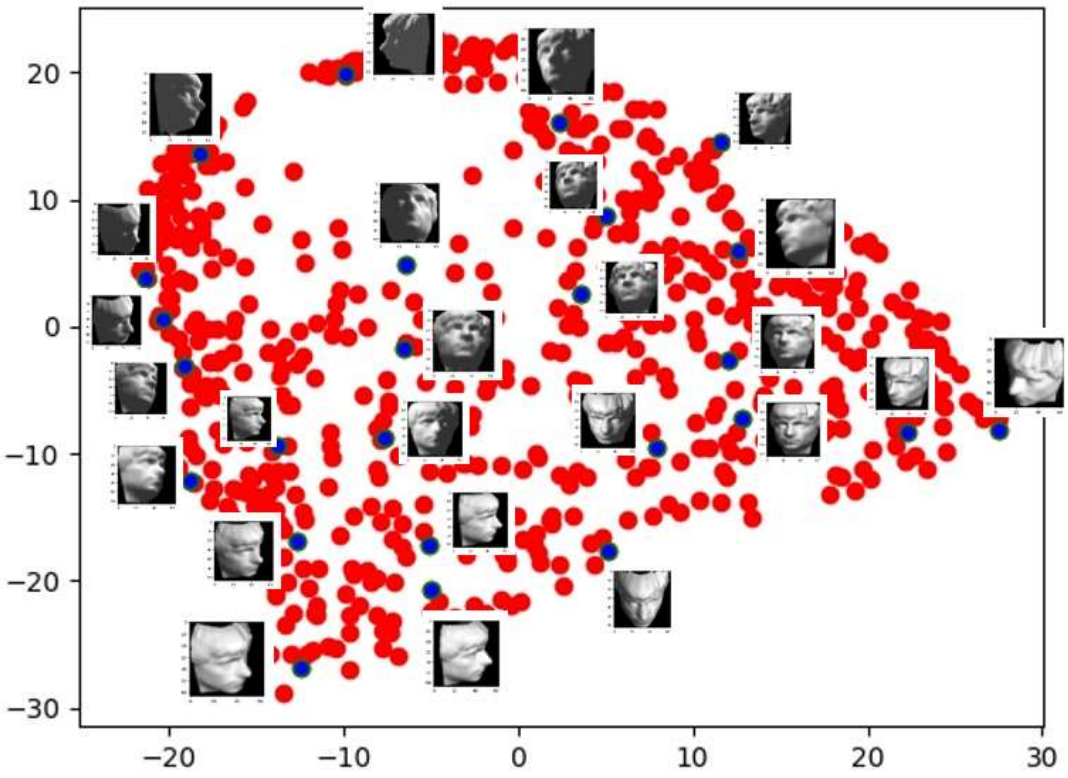
First, I tried the **strategy 1** to get the embedding image, which means I use the global ϵ and many nodes have 400-500 neighbors. The embedding is like this:



As you can see, the interpretation is very poor. That's because if I use the global ϵ, which will lead to many nodes have 400-500 neighbors, to some degree I am playing PCA in the analysis. If many nodes have many nodes(in this case, the total number of images is 698, so 500/698 is about 70%), it just break the original data space. For example, if the ϵ is too large, the distance between the two nodes will be the straight line, which is similar to PCA analysis.



So, I tried the **strategy 2**, which is to set the local ϵ for each node, and guarantees that each node has at least 100 neighbors. In this case, the original data space and the distances between them will be preserved greatly. The embedding image is shown as below:

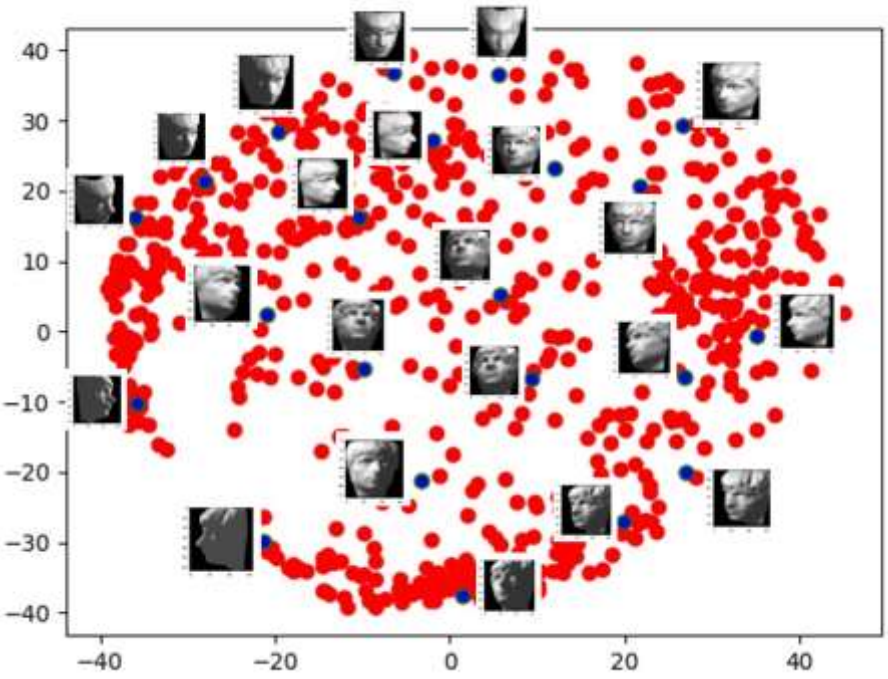


As you can see, this image is highly interpretable:

1. The horizontal axis, from negative -25 to 30, represents both the lighting direction and looking direction. If the value is negative, we will see the person from the left-hand side, and the left face will be lighted up. If the value is positive, we will see the person from the right-hand side, and the right face will be lighted up.
2. The vertical axis, from negative -30 to 25, represents the Up-down pose. If the value is negative, we will see the person from a higher position. If the value is positive, we will see the person in a lower position.
3. The images in the center are all the front view, which is corresponding to 1 and 2.

(c)

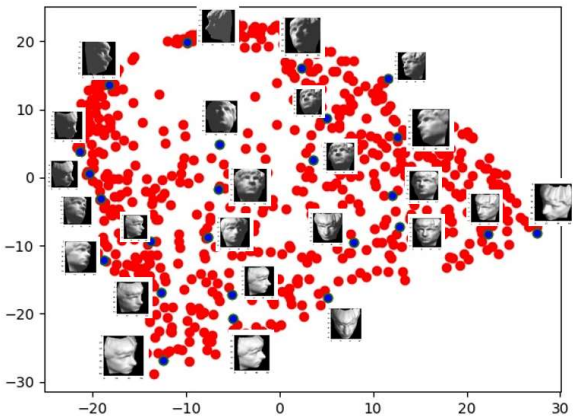
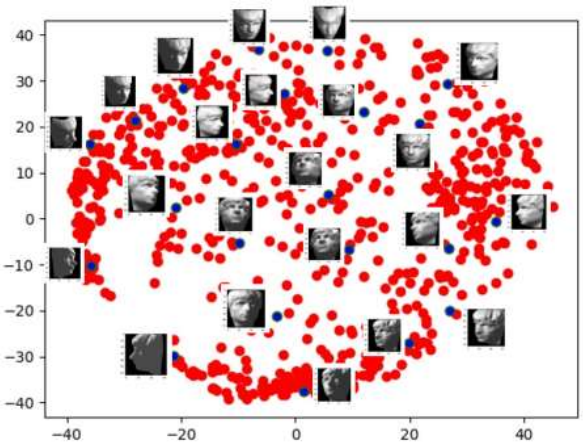
In part (c), I only use the strategy 2 and Manhattan distance to get the similarity matrix **A**. The result is shown as below:



As you can see, this image is highly interpretable:

1. The horizontal axis, from negative -42 to 48, represents both the looking direction. If the value is negative, we will see the person from the left-hand side. If the value is positive, we will see the person from the right-hand side.
2. The vertical axis, from negative -42 to 42, represents the Up-down pose. If the value is negative, we will see the person from a higher position. If the value is positive, we will see the person in a lower position.
3. The images in the center are all the front view, which is corresponding to 1 and 2.

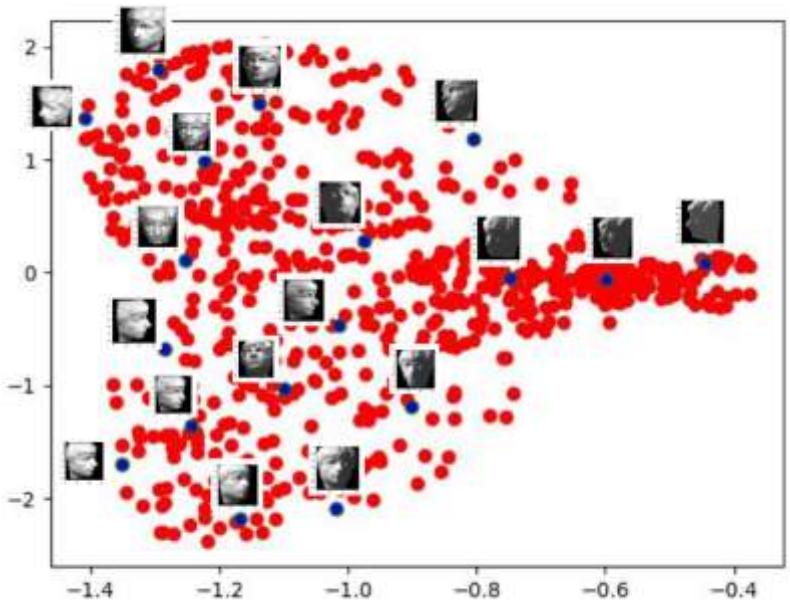
Compared the results from (b) and (c), we can conclude:

Euclidean distance	Manhattan distance
	

1. The shape of the scatter plots is similar with each other.
2. The interpretation of Euclidean distance is better than that of Manhattan distance. That is because the Euclidean distance will Magnify the differences by taking the square.
3. The range of horizontal axis and vertical axis is larger in Manhattan distance is larger than that in Euclidean distance. That is because the luminosity is between 0 and 1, so when taking the square of each element in Euclidean distance, the value will get smaller.

(d)

Taking the image luminosity as the features, I use the 4096 features with 698 samples to perform PCA, the embedding picture is shown as followed:



The result is much similar to the one in the lecture slides. Compared with the one in (b), the PCA embedding picture is not interpretable. I can't find any conclusion related to the axis. So the one in (b) is more meaningful.

The reason is that if you use PCA, you implicitly use the Euclidean distance to measure the similarity. For some complex 3D data space(or higher dimensional data space), PCA will largely break the distance in original space, just like the following picture. PCA will calculate the distance using the straight line, which is not true.

However, ISOMAP is a new way to project the original data distance to a lower dimensional space using **neighbors**. By setting neighbors and calculating the distance by cumulating the neighbor's distance, ISOMAP preserve the original data distance greatly. So the projection is more interpretable.