hw4

杨景兰 Yang Jinglan 12109069 9

1. (3 points) EM for mixtures of Bernoullis.
(1) Show that the M step for maximum likelihood estimation of a mixture of Bernoullis is given by

$$\mu_{kj} = \frac{\sum_i r_{ik} x_{ij}}{\sum_i r_{ik}}$$

Hint1: The distribution of a mixture of Bernoulli's is given by:

$$p(x_i|\mu_k) = \prod_{j=1}^{D} \mu_{kj}^{x_{ij}} (1-\mu_{kj})^{1-x_{ij}}$$

Hint2: μ_{kj} is given by :

$$\mu_{kj} = \arg\max_{\mu_{kj}} L(q;\mu) \quad (1)$$

Where $L(q;\mu)$ is the Auxiliary function (check ppg.17 in L16 slide).

Hint3: You can refer to (11.20)-(11.26) in Murphy's book

$p(X_i|\mu_k) = \prod_{j=1}^{D} \mu_{kj}^{x_{ij}} (1-\mu_{kj})^{1-x_{ij}}$

$\log p(X_i|\mu_k) = \sum_{j=1}^{D} [x_{ij} \log \mu_{kj} + (1-x_{ij}) \log(1-\mu_{kj})]$

The expected complete-data log-likelihood:

$Q(\mu, \mu^{old}) = \sum_{i=1}^{N} \sum_{k=1}^{K} r_{ik} \log p(X_i|\mu_k)$

$= \sum_{i=1}^{N} \sum_{k=1}^{K} r_{ik} \sum_{j=1}^{D} [x_{ij} \log \mu_{kj} + (1-x_{ij}) \log(1-\mu_{kj})]$

r_{ik} is the posterior probability of observation i belonging to mixture component k.

X_i is the i-th observation, μ_k is the vector of bernoulli parameters for component k.

μ^{old} is the previous estimate of the bernoulli parameter.

$$\frac{\partial Q(\mu, \mu^{old})}{\partial \mu_{kj}} = 0$$

$$\Rightarrow \sum_{i=1}^{N} r_{ik} \left(\frac{x_{ij}}{\mu_{kj}} - \frac{1-x_{ij}}{1-\mu_{kj}} \right) = \frac{\sum_{i=1}^{N} r_{ik} x_{ij}}{\mu_{kj}} - \frac{\sum_{i=1}^{N} r_{ik}(1-x_{ij})}{1-\mu_{kj}} = 0$$

$$\Rightarrow \mu_{kj} = \frac{\sum_{i=1}^{N} r_{ik} x_{ij}}{\sum_{i=1}^{N} r_{ik}}$$

(2) Show that the M step for MAP estimation of a mixture of Bernoullis with a $\beta(\alpha, \beta)$ prior is given by

$$\mu_{kj} = \frac{(\sum_i r_{ik} x_{ij}) + \alpha - 1}{(\sum_i r_{ik}) + \alpha + \beta - 2}$$

Hint4: $\beta(\alpha, \beta)$ prior is given by:

$$\beta(\alpha, \beta, \mu_{kj}) = A \mu_{kj}^{\alpha-1} (1-\mu_{kj})^{\beta-1} \quad \left(\because A = \frac{(\alpha+\beta-1)!}{(\alpha-1)!(\beta-1)!} \right)$$

Hint5: μ_{kj} is given by :

$$\mu_{kj} = \arg\max_{\mu_{kj}} L(q;\mu) + \sum_j \sum_k \log \beta(\alpha, \beta, \mu_{kj}) \quad (2)$$

$\mu_{kj} = \arg\max_{\mu_{kj}} L(q;\mu) + \sum_j \sum_k \log \beta(\alpha, \beta, \mu_{kj}) = \arg\max_{\mu_{kj}} Q(\mu, \mu^{old}) + \log \beta(\alpha, \beta)$.

$\log \beta(\alpha, \beta) = \log \prod_{k=1}^{K} \beta(\alpha, \beta, \mu_{kj})$

$= \log \prod_{k=1}^{K} A \mu_{kj}^{\alpha-1} (1-\mu_{kj})^{\beta-1}$

$= \sum_{k=1}^{K} [\log A + (\alpha-1) \log \mu_{kj} + (\beta-1) \log(1-\mu_{kj})]$

$$\frac{\partial \log \beta(\alpha, \beta)}{\partial \mu_{kj}} = \frac{\alpha-1}{\mu_{kj}} - \frac{\beta-1}{1-\mu_{kj}}$$

$$\Rightarrow \frac{\partial Q(\mu, \mu^{old})}{\partial \mu_{kj}} + \frac{\partial \log \beta(\alpha, \beta)}{\partial \mu_{kj}} = 0$$

$$\Rightarrow \frac{\sum_{i=1}^{N} r_{ik} x_{ij}}{\mu_{kj}} - \frac{\sum_{i=1}^{N} r_{ik}(1-x_{ij})}{1-\mu_{kj}} + \frac{\alpha-1}{\mu_{kj}} - \frac{\beta-1}{1-\mu_{kj}} = \frac{\sum_{i=1}^{N} r_{ik} x_{ij} + \alpha - 1}{\mu_{kj}} - \frac{\sum_{i=1}^{N} r_{ik}(1-x_{ij}) + \beta - 1}{1-\mu_{kj}} = 0$$

$$\Rightarrow \mu_{kj} = \frac{\sum_{i=1}^{N} r_{ik} x_{ij} + \alpha - 1}{\sum_{i=1}^{N} r_{ik} + \alpha + \beta - 2}$$

2. (2 points) Given the data set as shown in Figure 1 and assume that points $A1$, $A4$ and $A7$ are chosen to be the initialized cluster centers. The coordinates of the data points are:

$A1 = (2, 10)$, $A2 = (2, 5)$, $A3 = (8, 4)$, $A4 = (5, 8)$,
$A5 = (7, 5)$, $A6 = (6, 4)$, $A7 = (1, 2)$, $A8 = (4, 9)$.

(1) (1 points) Use the K-means algorithm and Euclidean distance to cluster the 8 data points into K = 3 clusters.
(2) (1 points) Consider the case that there exist 2 must links and 1 cannot link as shown in Figure 2, derive the 3 clusters now.

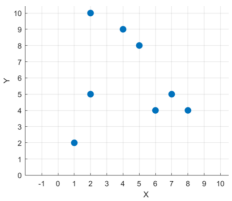


Figure 1: Training data set for K-means clustering

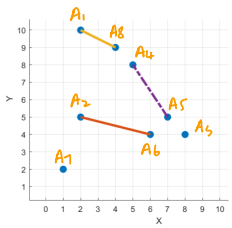


Figure 2: Training data set for constrained K-means clustering

(1) ① Iteration 1:

$d_{12}=5$, $d_{13}=6\sqrt{2}$, $d_{15}=5\sqrt{2}$, $d_{16}=2\sqrt{13}$, $d_{18}=\sqrt{5}$

$d_{42}=3\sqrt{2}$, $d_{43}=5$ ✓, $d_{45}=\sqrt{13}$ ✓, $d_{46}=\sqrt{17}$ ✓, $d_{48}=\sqrt{2}$ ✓

$d_{72}=\sqrt{10}$ ✓, $d_{73}=\sqrt{53}$, $d_{75}=6\sqrt{5}$, $d_{76}=\sqrt{29}$, $d_{78}=\sqrt{58}$

⇒ cluster 1 = $\{A_1\}$, cluster 2 = $\{A_3, A_4, A_5, A_6, A_8\}$, cluster 3 = $\{A_2, A_7\}$

center 1 = $(2, 10)$, center 2 = $\left(\frac{8+5+7+6+4}{5}, \frac{4+8+5+4+9}{5}\right) = (6, 6)$, center 3 = $\left(\frac{2+1}{2}, \frac{5+2}{2}\right) = \left(\frac{3}{2}, \frac{7}{2}\right)$

② Iteration 2:

$d_{12}=5$, $d_{13}=6\sqrt{2}$, $d_{14}=\sqrt{13}$, $d_{15}=5\sqrt{2}$, $d_{16}=2\sqrt{13}$, $d_{17}=\sqrt{65}$, $d_{18}=\sqrt{5}$ ✓

$d_{22}=\sqrt{17}$, $d_{23}=2\sqrt{2}$ ✓, $d_{24}=\sqrt{5}$ ✓, $d_{25}=\sqrt{2}$ ✓, $d_{26}=2$ ✓, $d_{27}=\sqrt{41}$, $d_{28}=\sqrt{13}$

$d_{32}=\sqrt{\frac{5}{2}}$ ✓, $d_{33}=\sqrt{\frac{85}{2}}$, $d_{34}=\sqrt{\frac{65}{2}}$, $d_{35}=\sqrt{\frac{65}{2}}$, $d_{36}=\sqrt{\frac{41}{2}}$, $d_{37}=\sqrt{\frac{5}{2}}$ ✓, $d_{38}=\sqrt{\frac{73}{2}}$

cluster 1 = $\{A_1, A_8\}$, cluster 2 = $\{A_3, A_4, A_5, A_6\}$, cluster 3 = $\{A_2, A_7\}$

center 1 = $(3, 9.5)$, center 2 = $\left(\frac{8+5+7+6}{4}, \frac{4+8+5+4}{4}\right) = (6.5, 5.25)$, center 3 = $(1.5, 3.5)$

③ Iteration 3:

$d_{11}=\sqrt{1.25}$ ✓, $d_{12}=\sqrt{21.25}$, $d_{13}=\sqrt{55.25}$, $d_{14}=\sqrt{6.25}$ ✓, $d_{15}=\sqrt{36.25}$, $d_{16}=\sqrt{39.25}$, $d_{17}=\sqrt{60.25}$, $d_{18}=\sqrt{1.25}$ ✓

$d_{21}=\sqrt{42.8125}$, $d_{22}=\sqrt{20.3125}$, $d_{23}=\sqrt{3.8125}$ ✓, $d_{24}=\sqrt{9.8125}$, $d_{25}=\sqrt{0.3125}$ ✓, $d_{26}=\sqrt{1.8125}$ ✓, $d_{27}=\sqrt{40.8125}$, $d_{28}=\sqrt{20.3125}$

$d_{31}=\sqrt{42.5}$, $d_{32}=\sqrt{2.5}$ ✓, $d_{33}=\sqrt{42.5}$, $d_{34}=\sqrt{32.5}$, $d_{35}=\sqrt{32.5}$, $d_{36}=\sqrt{20.5}$, $d_{37}=\sqrt{2.5}$ ✓, $d_{38}=\sqrt{36.5}$

cluster 1 = $\{A_1, A_4, A_8\}$, cluster 2 = $\{A_3, A_5, A_6\}$, cluster 3 = $\{A_2, A_7\}$

center 1 = $\left(\frac{11}{3}, 9\right)$, center 2 = $\left(7, \frac{13}{3}\right)$, center 3 = $(1.5, 3.5)$

④ Iteration 4:

$d_{11}=\sqrt{1.44}$ ✓ $d_{12}=\sqrt{18.78}$ $d_{13}=\sqrt{43.78}$ $d_{14}=\sqrt{2.78}$ ✓ $d_{15}=\sqrt{27.11}$, $d_{16}=\sqrt{30.44}$, $d_{17}=\sqrt{56.11}$, $d_{18}=\sqrt{0.11}$ ✓

$d_{21}=\sqrt{57.11}$ $d_{22}=\sqrt{25.44}$ $d_{23}=\sqrt{1.11}$ ✓ $d_{24}=\sqrt{17.44}$ $d_{25}=\sqrt{0.44}$ ✓, $d_{26}=\sqrt{1.11}$ ✓, $d_{27}=\sqrt{41.44}$, $d_{28}=\sqrt{30.77}$

$d_{31}=\sqrt{42.5}$, $d_{32}=\sqrt{2.5}$ ✓, $d_{33}=\sqrt{42.5}$, $d_{34}=\sqrt{32.5}$, $d_{35}=\sqrt{32.5}$, $d_{36}=\sqrt{20.5}$, $d_{37}=\sqrt{2.5}$ ✓, $d_{38}=\sqrt{36.5}$

cluster 1 = $\{A_1, A_4, A_8\}$, cluster 2 = $\{A_3, A_5, A_6\}$, cluster 3 = $\{A_2, A_7\}$

⇒ The clusters do not change after iteration 3.

Hence, the final clusters = cluster 1 = $\{A_1, A_4, A_8\}$, cluster 2 = $\{A_3, A_5, A_6\}$, cluster 3 = $\{A_2, A_7\}$

(2) ① Iteration 1 = cluster 1 = $\{A_1, A_4, A_8\}$, cluster 2 = $\{A_3, A_5, A_6\}$, cluster 3 = $\{A_2, A_7\}$

center 1 = $\left(\frac{11}{3}, 9\right)$, center 2 = $\left(7, \frac{13}{3}\right)$, center 3 = $(1.5, 3.5)$

	1	2	3	
A_1	✓ 1.44	57.11	42.5	
A_2	18.78	25.44	2.5 ✓	
A_3	43.78	✓ 1.11	42.5	
A_4	✓ 2.78	17.44	32.5	⇒ $\{A_1, A_4, A_8\}$ $\{A_6, A_5\}$ $\{A_2, A_6, A_7\}$
A_5	27.11	✓ 0.44	32.5	
A_6	30.44	1.11	20.5 ✓	center 1 = $(\frac{11}{3}, 9)$, center 2 = $(\frac{15}{2}, \frac{9}{2})$, center 3 = $(3, \frac{11}{3})$
A_7	56.11	41.44	2.5 ✓	
A_8	✓ 0.11	30.77	36.5	

② Iteration 2 = center 1 = $(\frac{11}{3}, 9)$, center 2 = $(\frac{15}{2}, \frac{9}{2})$ = $(7.5, 4.5)$, center 3 = $(3, \frac{11}{3})$ = $(3, 3.33)$

	1	2	3

A_1	✓1.44	60.5	41.11	
A_2	18.78	30.5	2.78✓	
A_3	43.78	0.5✓	25.45	
A_4	✓2.78	18.5	25.81	⇒ $\{A_1, A_4, A_8\}$ $\{A_5, A_5\}$ $\{A_2, A_6, A_7\}$
A_5	27.11	0.5✓	18.79	
A_6	30.64	2.5	9.45✓	⇒ The clusters do not change.
A_7	56.11	48.5	5.77✓	
A_8	✓0.1	32.5	33.15	

Hence, the final clusters = cluster 1 = $\{A_1, A_4, A_8\}$, cluster 2 = $\{A_3, A_5\}$, cluster 3 = $\{A_2, A_6, A_7\}$

3. (2 point) Consider the following 10 data points: $X = \{(7,4,3),(4,1,8),(6,3,5),(8,6,1),(8,5,7),(7,2,9),(5,3,3),(9,5,8),(7,4,5),(8,2,2)\}$. Calculate the projection of each data onto a two-dimensional subspace(i.e.K=2) using PCA. You could use python or matlab or online calculator to obtain eigenvectors and eigenvalues. You should show each step of deriving the result.(You don't need to show the calculation)

$$X = \begin{bmatrix} 7&4&3\\ 4&1&8\\ 6&3&5\\ 8&6&1\\ 8&5&7\\ 7&2&9\\ 5&3&3\\ 9&5&8\\ 7&4&5\\ 8&2&2 \end{bmatrix} \Rightarrow X\_mean = (6.9\ \ 3.5\ \ 5.1) \Rightarrow X\_centered = X - X\_mean = \begin{bmatrix} 0.1&0.5&-2.1\\ -2.9&-2.5&2.9\\ -0.9&-0.5&-0.1\\ 1.1&2.5&-4.1\\ 1.1&1.5&1.9\\ 0.1&-1.5&3.9\\ -1.9&-0.5&-2.1\\ 2.1&1.5&2.9\\ 0.1&0.5&-0.1\\ 1.1&-1.5&-3.1 \end{bmatrix}$$

$$\text{covariance matrix} = \frac{1}{10-1}(X\_centered)^T(X\_centered) = \frac{1}{9}\begin{bmatrix} 0.1&-2.9&-0.9&1.1&1.1&0.1&-1.9&2.1&0.1&1.1\\ 0.5&-2.5&-0.5&2.5&1.5&-1.5&-0.5&1.5&0.5&-1.5\\ -2.1&2.9&-0.1&-4.1&1.9&3.9&-2.1&2.9&-0.1&-3.1 \end{bmatrix} \begin{bmatrix} 0.1&0.5&-2.1\\ -2.9&-2.5&2.9\\ -0.9&-0.5&-0.1\\ 1.1&2.5&-4.1\\ 1.1&1.5&1.9\\ 0.1&-1.5&3.9\\ -1.9&-0.5&-2.1\\ 2.1&1.5&2.9\\ 0.1&0.5&-0.1\\ 1.1&-1.5&-3.1 \end{bmatrix}$$

$$= \begin{bmatrix} 2.32&1.61&-0.43\\ 1.61&2.50&-1.28\\ -0.43&-1.28&7.88 \end{bmatrix}$$

eigenValues = [0.74992815, 3.67612927, 8.27394258]

$$\text{eigenVectors} = \begin{bmatrix} -0.70177243&0.69903712&-0.13758708\\ 0.70745703&0.66088917&-0.25045969\\ 0.08416157&0.27307986&0.95830278 \end{bmatrix}$$

Sort the eigenVectors based on their corresponding eigenValues:

$$\text{eigenVectors}' = \begin{bmatrix} -0.13758708&0.69903712&-0.70177243\\ -0.25045969&0.66088917&0.70745703\\ 0.95830278&0.27307986&0.08416157 \end{bmatrix}$$

↳ choose these 2 columns for PCA.

$$\text{Transformed} = \begin{bmatrix} 0.1&0.5&-2.1\\ -2.9&-2.5&2.9\\ -0.9&-0.5&-0.1\\ 1.1&2.5&-4.1\\ 1.1&1.5&1.9\\ 0.1&-1.5&3.9\\ -1.9&-0.5&-2.1\\ 2.1&1.5&2.9\\ 0.1&0.5&-0.1\\ 1.1&-1.5&-3.1 \end{bmatrix} \begin{bmatrix} -0.13758708&0.69903712\\ -0.25045969&0.66088917\\ 0.95830278&0.27307986 \end{bmatrix} = \begin{bmatrix} -2.15142276&-0.17311941\\ 3.80418259&-2.88749898\\ 0.15321328&-0.98688598\\ -4.70651185&1.30153634\\ 1.29375788&2.27912632\\ 4.09931333&0.14358214\\ -1.62582148&-2.23208282\\ 2.11448986&3.25124433\\ -0.23481772&0.47304051\\ -2.76637697&-1.06894049 \end{bmatrix}$$

$$\text{Hence, the final projection} = \begin{bmatrix} -2.15142276&-0.17311941\\ 3.80418259&-2.88749898\\ 0.15321328&-0.98688598\\ -4.70651185&1.30153634\\ 1.29375788&2.27912632\\ 4.09931333&0.14358214\\ -1.62582148&-2.23208282\\ 2.11448986&3.25124433\\ -0.23481772&0.47304051\\ -2.76637697&-1.06894049 \end{bmatrix}$$

DDA3020 Assignment 4 report (coding part)

1. Data preprocessing

1.1 check whether the dataset has "NaN" or not use df.info()

→I have adjusted the data by hand, since some data are mismatch its position

→this dataset is clear—without "NaN"

1.2 Normalization

Before we do PCA, we have to center and scale the data.

After centering, the average value for each data will be 0, and after scaling, the standard deviation for the values for each data will be 1.

$$standard\ score = \frac{x - \mu}{\sigma}$$

2. PCA

The main steps involved in performing PCA are:

(1). Standardize the data: just what we have done in 1.2 Normalization.

(2). Compute the covariance matrix: $Covariance = \frac{1}{n}XX^T$ or np.cov(X.T).

(3). Compute the eigenvectors and eigenvalues of the covariance matrix: The eigenvectors and eigenvalues represent the new coordinate system and the amount of variance explained by each component, respectively. The eigenvectors are ordered by their corresponding eigenvalues in descending order. (np.linalg.eig(cov_mat))

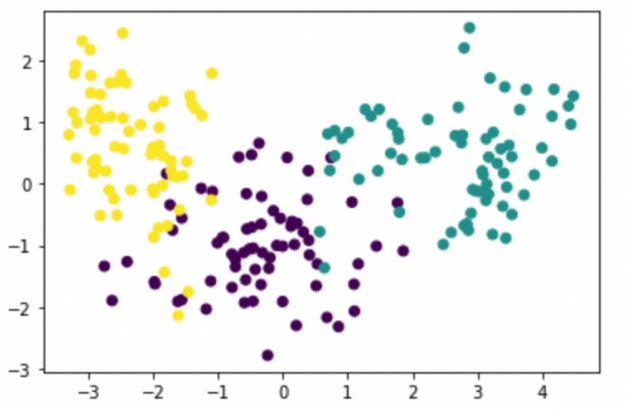
(4). Select the number of principal components: The number of principal components to retain depends on the amount of variance explained by each component. A common rule of thumb is to retain enough components to explain at least 70-80% of the variance in the data. Here, since we project each data point onto a two-dimensional subspace, n_components is set to 2.

(5). Transform the data: The final step is to project the data onto the new coordinate system defined by the eigenvectors. This creates a new set of variables, called principal components, that are uncorrelated and ordered by the amount of variance they explain. (transformed_data= np.dot(X, top_eigenvectors.T))

The final projection: (just list some of the projections)

```
[[ 3.17047052e-01 -7.83669015e-01]
 [-3.38617773e-03 -1.91321367e+00]
 [-4.59443341e-01 -1.90722525e+00]
 [-5.91935651e-01 -1.93106922e+00]
 [ 1.10291005e+00 -2.06809003e+00]
 [-3.36815150e-01 -1.63691626e+00]
 [-1.45602498e-01 -4.36414570e-01]
 [-4.25439933e-01 -1.39015330e+00]
 [ 1.76718434e+00 -3.07528830e-01]
 [ 1.44119415e+00 -1.01106830e+00]
 [ 6.71107449e-02  4.29312461e-01]
 [-5.72683493e-01 -1.55824408e+00]
 [-9.31414541e-01 -8.73925640e-01]
 [-9.14882766e-01 -8.71940034e-01]
 [-1.00836149e+00 -9.58530316e-01]
 [-5.15522337e-01 -1.05872649e+00]
 [-7.84530515e-01 -1.14197655e+00]
 [ 6.79029059e-01 -2.17036180e+00]
 [-2.34416607e-01 -2.78779387e+00]
 [-1.73767215e+00 -3.40140568e-01]
 [-5.61482567e-01 -1.58659628e-01]
 [-4.77524312e-01 -7.04746626e-01]
 [ 8.57383186e-01 -2.32117455e+00]
 [-1.97851961e+00 -1.59076489e+00]
 [-1.57527098e-03 -1.01326624e+00]
 [ 1.09652119e+00 -1.63066030e+00]
 [-1.55728246e+00 -5.55134014e-01]
 [-1.69804238e+00 -7.50450169e-01]
 [-4.56618638e-01 -1.04421405e+00]
 [-1.08377612e+00 -1.22309883e-01]
```

Visualize the projection as following:



3. K-means

(1). Initialize the cluster centroids: The algorithm starts by randomly selecting K data points from the dataset as the initial cluster centroids.

(2). Assign each data point to the nearest centroid: For each data point, compute the distance to each centroid and assign it to the nearest centroid.

(3). Update the cluster centroids: Compute the mean of all data points assigned to each cluster and set the centroid to the mean.

(4). Repeat steps 2 and 3 until convergence: The above two steps are repeated until the centroids no longer change, or the maximum number of iterations is reached.

Labels from K-means:

```
[0 0 0 0 0 0 0 0 2 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0
 2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 1 0 0 0 0 0 1 2 2 2 2
 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
 2 2 2 2 2 2 2 2 2 2 2 2 0 2 2 2 2 2 2 2 0 2 2 0 2 0 0 2 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 0 1 0 1 1 1 1 1 1 1 1]
```

4. Silhouette Coefficient

(1). Calculate the average distance (a(i)) between i and all other data points in its own cluster.

(2). Calculate the average distance (b(i)) between i and all data points in the nearest cluster (i.e., the cluster with the smallest distance to i).

(3). Calculate the Silhouette Coefficient: $s(i) = \frac{b(i)-a(i)}{\max(a(i),b(i))}$

Result: S= 0.5000858694646493

5. Rand Index

$$RI = \frac{a+b}{a+b+c+d} = \frac{a+b}{\frac{n(n-1)}{2}}$$

A (true positive): The number of pairs of elements in S that are in the same subset in True_categories and in the same subset in Predict_categories

B (true negative): The number of pairs of elements in S that are in the different subset in True_categories and in the different subset in Predict_categories

C (false positive): The number of pairs of elements in S that are in the same subset in True_categories and in the different subset in Predict_categories

D (false negative): The number of pairs of elements in S that are in the different subset in

True_categories and in the same subset in Predict_categories
Result: RI= 0.8890407837776259