Yohandi (120040025)
DDA3020 Assignment 4

1(1) In M-Step for Gaussian Mixture Models,

$$\theta^{new} = \underset{\theta}{\arg\max} \sum_{n=1}^{N} \sum_{k=1}^{K} \gamma_k^{(n)} \left(\log \pi_k + \log N(x^{(n)}; \mu_k, \Sigma_k) \right)$$

since the mixture is defined as Bernoullis' distribution,

$N(x^{(n)}; \mu_k, \Sigma_k)$ is ~~[illegible]~~ replaced by $\prod_j (\mu_{kj}^{x_j} \cdot (1-\mu_{kj})^{1-x_j})$

also, since the distribution is discrete and hardly assigned,

we use r_{ik} instead of $\gamma_k^{(i)}$

$$\mu^{new} = \underset{\mu}{\arg\max} \sum_{i=1}^{N} \sum_{k=1}^{K} r_{ik} \left(\log \pi_k + \log \left(\prod_j \mu_{kj}^{x_j} \cdot (1-\mu_{kj})^{1-x_j} \right) \right)$$

$$= \underset{\mu}{\arg\max} \sum_i \sum_k r_{ik} \left(\log \pi_k + \sum_j \log (\mu_{kj}^{x_j} \cdot (1-\mu_{kj})^{1-x_j}) \right)$$

$$= \underset{\mu}{\arg\max} \sum_i \underbrace{\sum_k r_{ik} \left(\log \pi_k + \sum_j \left[x_j \cdot \log(\mu_{kj}) + (1-x_j) \log(1-\mu_{kj}) \right] \right.}_{f}$$

$$\frac{\partial f}{\partial \mu_{kj}} = \sum_i r_{ik} \left(x_{ij} \cdot \frac{\partial (\log(\mu_{kj}))}{\partial \mu_{kj}} + (1-x_{ij}) \cdot \frac{\partial \log(1-\mu_{kj})}{\partial \mu_{kj}} \right)$$

$$= \sum_i r_{ik} \left(x_{ij} \cdot \frac{1}{\mu_{kj}} + (1-x_{ij})(-1) \cdot \frac{1}{1-\mu_{kj}} \right)$$

$$= \sum_i r_{ik} \left(\frac{x_{ij}}{\mu_{kj}} - \frac{(1-x_{ij})}{1-\mu_{kj}} \right)$$

set $\frac{\partial f}{\partial \mu_{kj}} = 0 \Rightarrow \sum_i r_{ik} \left(\frac{x_{ij}}{\mu_{kj}} - \frac{(1-x_{ij})}{1-\mu_{kj}} \right) = 0$

$$\Rightarrow \sum_i r_{ik} \left(\frac{x_{ij} - \cancel{x_{ij}\mu_{kj}} - \mu_{kj} + \cancel{x_{ij}\mu_{kj}}}{\mu_{kj}(1-\mu_{kj})} \right) = 0$$

since hard assignment implies $r_{ik} \in \{0, 1\}$

$$\Rightarrow \sum_i r_{ik} (x_{ij} - \mu_{kj}) = 0$$

$$\Rightarrow \sum_i r_{ik} x_{ij} = \underbrace{\left(\sum_i r_{ik} \right)}_{\text{constant}} \cdot \mu_{kj}$$

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

(2) Similar with (1); however, we want to include α and β as priors,

$$\mu^{new} = \underset{\mu}{\arg\max} \underbrace{\sum_i \sum_k r_{ik} \left(\log \pi_k + \sum_j \left[x_j \cdot \log(\mu_{kj}) + (1-x_j) \log(1-\mu_{kj}) \right] \right) + \sum_k \sum_j \left[(\alpha-1) \log(\mu_{kj}) + (\beta-1) \log(1-\mu_{kj}) \right]}_{g}$$

turn paper

$$\frac{\partial g}{\partial \mu_{kj}} = \underbrace{\sum_i \left(r_{ik}\left(\frac{x_{ij}}{\mu_{kj}} - \frac{(1-x_{ij})}{(1-\mu_{kj})}\right)\right)}_{\text{similar with previous part}} + \sum_k \sum_j \left(\frac{\alpha-1}{\mu_{kj}} - \frac{\beta-1}{1-\mu_{kj}}\right)$$

$$= \frac{\alpha-1}{\mu_{kj}} - \frac{\beta-1}{1-\mu_{kj}}$$ since we only focus on specific k & j

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

set $\frac{\partial g}{\partial \mu_{kj}} = 0 \Rightarrow \sum_i r_{ik}\left(\frac{x_{ij} - \cancel{x_{ij}\mu_{kj}} - \mu_{kj} + \cancel{x_{ij}\mu_{kj}}}{\mu_{kj}(1-\mu_{kj})}\right) + \frac{\alpha - \alpha\mu_{kj} - 1 + \mu_{kj} - \beta\mu_{kj} + \mu_{kj}}{\mu_{kj}(1-\mu_{kj})} = 0$

$$\Rightarrow \sum_i r_{ik} x_{ij} - \underbrace{\sum_i r_{ik}}_{\text{constant}} \mu_{kj} + \mu_{kj}(-\alpha-\beta+2) + (\alpha-1) = 0$$

$$\Rightarrow \left(\sum_i r_{ik} x_{ij}\right) + \alpha - 1 = \mu_{kj}\left[\left(\sum_i r_{ik}\right) + \alpha + \beta - 2\right]$$

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

2. First, we have:

$$AUC = \frac{1}{m^+ m^-} \sum_{i=1}^{m^+} \sum_{j=1}^{m^-} u(e_{ij})$$

$$= \frac{1}{m^+ m^-} \sum_{i=1}^{m^+} \sum_{j=1}^{m^-} u(g(x_i^+) - g(x_j^-))$$

- $\rightarrow$ $u(g(x_i^+) - g(x_j^-))$ results 1 when $g(x_i^+) - g(x_j^-) > 0 \Rightarrow g(x_i^+) > g(x_j^-)$
- $\rightarrow$ the number of j that fulfills the previous claim is equivalent with rank i excluding all positive samples (so far, i)

$$\Rightarrow AUC = \frac{1}{m^+ m^-} \sum_{i=1}^{m^+} (\text{rank}_i - i)$$

$$= \frac{\sum_{i=1}^{m^+} \text{rank}_i - \sum_{i=1}^{m^+} i}{m^+ m^-}$$

$$= \frac{\sum_{i=1}^{m^+} \text{rank}_i - \frac{m^+(m^+ + 1)}{2}}{m^+ m^-}$$

3. $X = \begin{bmatrix} 2 & 0 & 1 & -3 & -2 \\ 0 & 2 & -3 & -3 & -2 \\ & & \vdots & & \\ -2 & 3 & -3 & 3 & 2 \\ -3 & 2 & 0 & -1 & -2 \end{bmatrix}^T$

$\mu = [-0.4 \;\; 0.8 \;\; 0.2 \;\; 0.2 \;\; -1.3]^T$

define a matrix M as $\begin{bmatrix} \mu \\ \mu \\ \vdots \\ \mu \end{bmatrix}^T$

Then, we have:

$$\Sigma = \frac{(X-M)(X-M)^T}{N-1}$$

$$= \frac{\begin{bmatrix} 2.4 & 0.4 & \cdots & -2.6 \\ -0.8 & 1.2 & & 1.2 \\ 0.8 & -3.2 & & -0.2 \\ -3.2 & -3.2 & & -1.2 \\ -0.7 & 0.7 & & -0.7 \end{bmatrix} \begin{bmatrix} 2.4 & 0.4 & & -2.6 \\ -0.8 & 1.2 & & 1.2 \\ 0.8 & -3.2 & \cdots & -0.2 \\ -3.2 & -3.2 & & -1.2 \\ -0.7 & 0.7 & & -0.7 \end{bmatrix}^T}{9}$$

$$= \begin{bmatrix} 3.37\bar{7} & 0.9\bar{1}\bar{1} & & -0.13\bar{3} \\ 0.91\bar{1} & 4.17\bar{7} & & 1.15\bar{5} \\ -0.02\bar{2} & -2.4 & \cdots & -0.93\bar{3} \\ -0.9\bar{1}\bar{1} & 1.15\bar{5} & & 1.28\bar{8} \\ -0.13\bar{3} & 1.15\bar{5} & & 2.45\bar{5} \end{bmatrix}$$

by Python, the corresponding eigenvalues and eigenvectors to the above Σ are:

→ eigenvalues = [0.9146, 1.7003, 3.4179, 6.5896, 7.5220]

→ eigenvectors = [... ,

[-0.3314, -0.1573, 0.1438, -0.6462, 0.6536],

[0.1696, 0.9104, -0.1105, 0.0303, 0.3595]]

we choose two eigenvectors that correspond to the two highest eigenvalues for PCA, then the representation is given as:

$Z = U^T(X-M)$ where U is a matrix of the chosen 2 eigen vectors

$$\Rightarrow Z = \begin{bmatrix} -0.7584 & 1.1654 & 1.0747 & -0.0668 & 0.2811 & \cdots \\ 1.0558 & 0.8287 & -2.8046 & -0.3970 & 2.0555 & \end{bmatrix}$$

representation for data point [0]   representation for data point [1]   ...   .

$$\cdots \begin{matrix} 2.3152 & 3.3565 & -3.1863 & -4.4972 & 0.3856 \\ -2.2887 & 0.0714 & -6.0370 & 1.5538 & 0.9627 \end{matrix} \Bigg]^T$$

...   .   representation for data point [9]

Programming Question

Yohandi [SID: 120040025]

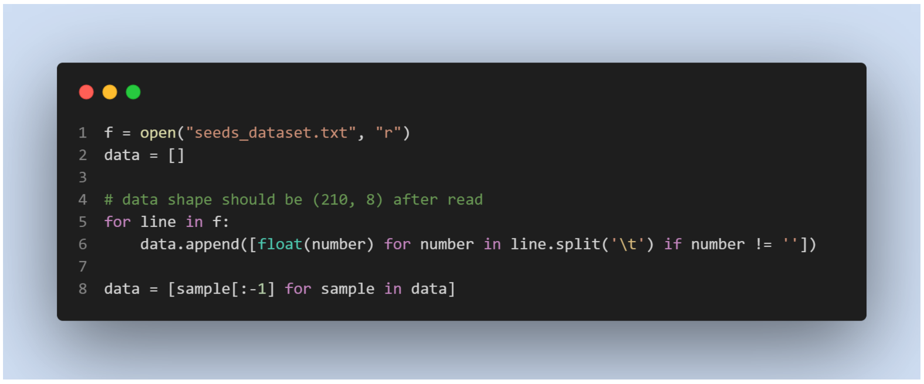
K-means and GMM-EM Algorithms Implementation

In this task, we are asked to implement both K-means and GMM-EM algorithms from scratch. In other words, no third-party or off-the-shelf package or library are allowed. The algorithms are expected to cluster the UCI seed dataset, which can be downloaded from https://archive.ics.uci.edu/ml/datasets/seeds. The number of clusters is set as 3.

K-means

The k-means clustering technique is a well-known unsupervised machine learning algorithm. Each data point is assigned to the cluster with the closest mean. At each iteration, the algorithm attempts to minimize the sum of squared distances between data points and their corresponding cluster means.

The program for the k-means algorithm starts by reading the given data from a file named `seeds_dataset.txt`. Then, the every elements in the data is assigned with float. Lastly, the last column in the data, which is the label class, is removed.



```python
f = open("seeds_dataset.txt", "r")
data = []

# data shape should be (210, 8) after read
for line in f:
    data.append([float(number) for number in line.split('\t') if number != ''])

data = [sample[:-1] for sample in data]
```

In the next part, the program initializes some necessary states: cluster centers position, initial assignments, etc. For the cluster centers, depending on the value of `is_random`, the lists are filled with k data points (fixed if `is_random` is false). The assignments, however, are filled with -1 to denote that all points are initially not assigned to any clusters.

Then in the last part, the program performs a loop that will stop only if the objective value is converged to some extents. The loop performs two main steps: E-step, and M-step.

In the first phase, E-step (expectation step), each data point is assigned to the cluster with the closest mean (center). Specifically, the algorithm computes the distance to the mean of each cluster for each data point and assigns each data point to the cluster with the shortest distance. Based on the present estimations of the cluster means, this phase estimates the membership of each data point inside each cluster. After deriving $\min_r \sum_i^n \sum_k^K r_{ik}(x_i - c_k)^2$ subject to $r \in \{0,1\}^{n\times K}$ and $\sum_k^K r_{ik} = 1$, it can be found that the solution is to set a variable $k^* = \arg\min\{(x_i - c_k)^2\}_{k=1}^K$ and set $r_{ik^*} = 1$.

The second phase, M-step (maximization step), includes updating the cluster mean estimations based on the assignments established in E-step. In particular, the new mean of each cluster is derived as the average of all data points assigned to that cluster. This phase maximizes the objective function that the k-means algorithm is optimizing, which is the sum of the squared distances between each data point and its assigned cluster mean. After deriving $\min_c \sum_i^n \sum_k^K r_{ik}(x_i - c_k)^2$, it can be found that $c_k = \frac{\sum_i^n r_{ik}x_i}{\sum_i^n r_{ik}}$ fulfills the solution for every $k \in K$.

GMM-EM

The GMM-EM technique is a probabilistic model that assumes each cluster's data points are generated from a Gaussian distribution. The program iteratively estimates the Gaussian distribution parameters and allocates data points to the cluster with the closest mean. It also can handle complex data distributions, including multimodal distributions, as well as missing data values. The GMM-EM algorithm permits soft clustering, which means that data points can partially belong to different clusters as opposed to being strictly assigned to a single cluster.

Similar to k-means implementation, the program for the GMM-EM algorithm starts by reading the given data from a file named `seeds_dataset.txt`. Then, the every elements in the data is assigned with float. Lastly, the last column in the data, which is the label class, is removed. Moreover, the initialize part is also the same as the previous implementation.

In the last part, the program performs a loop that will stop only if the log likelihood is converged to some extents. The loop performs two main steps: E-step, and M-step.

In the first phase, the E-step (expectation step) estimates the membership of each data point in each cluster based on the present estimates of cluster means. Specifically, the algorithm calculates, for each data point, the probability that the data point belongs to each cluster, given the current estimates of the cluster characteristics (such as the mean and variance). Then, the data point is assigned to the cluster with the highest probability. We obtain the responsibilities formula as: $\gamma_{ki} = \frac{\pi_k \mathcal{N}(x_i|\mu_k,\sigma_k)}{\sum_j^K \pi_j \mathcal{N}(x_i|\mu_j,\sigma_j)}$.

In the second phase, the M-step (maximization step) entails updating the cluster parameter estimations based on the E-assignments. step's In particular, the updated estimates of the cluster parameters (such as the mean and variance) are generated as the weighted average of all the data points assigned to that cluster, with the weights being the estimated probability of membership in the E-step. The objective function being optimized by the GMM-EM algorithm, this phase maximizes the likelihood of the data given the present estimates of the cluster characteristics. We also obtain the adjusted other parameters as: $\mu_k = \frac{1}{N_k}\sum_i^N \gamma_{ki} x_i$, $\sigma_k = \frac{1}{N_k}\sum_i^N \gamma_k^{(i)}(x^{(i)} - \mu_k)(x^{(i)} - \mu_k)^T$, and $\pi_k = \frac{N_k}{N}$.

Evaluation Metrics

As the presentation result of this report, we are asked to implement two evaluation metrics: silhouette coefficient, and rand index. Of course, no additional library is allowed.

Silhouette Coefficient

In short, silhouette coefficient quantifies an object's similarity to its own cluster relative to other clusters. It is a measurement of a cluster's cohesion. A high silhouette coefficient suggests that an object is well-suited to its own cluster but poorly matched to surrounding clusters. A value of 0 shows that the object is equally matched to both its own cluster and the cluster that is immediately adjacent to it.

To compute the silhouette coefficient for an object, you must first calculate the average distance between that object and all other objects in its own cluster (a) and then calculate the average distance between that object and all other objects in the cluster that is closest to it (b). The silhouette coefficient is then computed by $\frac{(b-a)}{\max(a,b)}$.

Rand Index

The Rand index measures the degree of similarity between two data clusterings. It is a measure of the degree of agreement between the two clusterings, with 1 indicating total agreement and 0 indicating no agreement. The measurement is defined as $\frac{a+b}{\frac{n\times(n-1)}{2}}$, where a denotes the number of pairs of objects that are in the same cluster in both clusterings and b denotes the number of pairs of objects that are in different clusters in both clusterings.

Results

K-means

Picking up the lowest obtained objective value from 30 trial(s), we have the cluster sizes as $[57, 65, 88]$ with 4 iterations. For the Silhouette Coefficient for K-means, we obtain a value of 0.48596. For the Rand Index for K-means, we obtain a value of 0.63805.

In randomized cases of initial clusters position, we obtain: standard deviation of Silhouette Coefficient for K-means is 0.03817, and standard deviation of Rand Index for K-means is 0.04788.

GMM-EM

From a total of 30 iterations, we have the approximation of labeled data sizes as $[67, 60, 83]$. For the Silhouette Coefficient for GMM-EM, we obtain a value of 0.67058. For the Rand Index for GMM-EM, we obtain a value of 0.88649.

In randomized cases of initial mean and covariance values, we obtain: standard deviation of Silhouette Coefficient for GMM-EM is 0.16653, and standard deviation of Rand Index for GMM-EM is 0.10332.