Name : Vedadnya Jadhav

CWID : 10450603

Problem 1 (25pt): [K-means] Implement the K-means algorithm. Note that you cannot directly call the built-in kmeans functions.

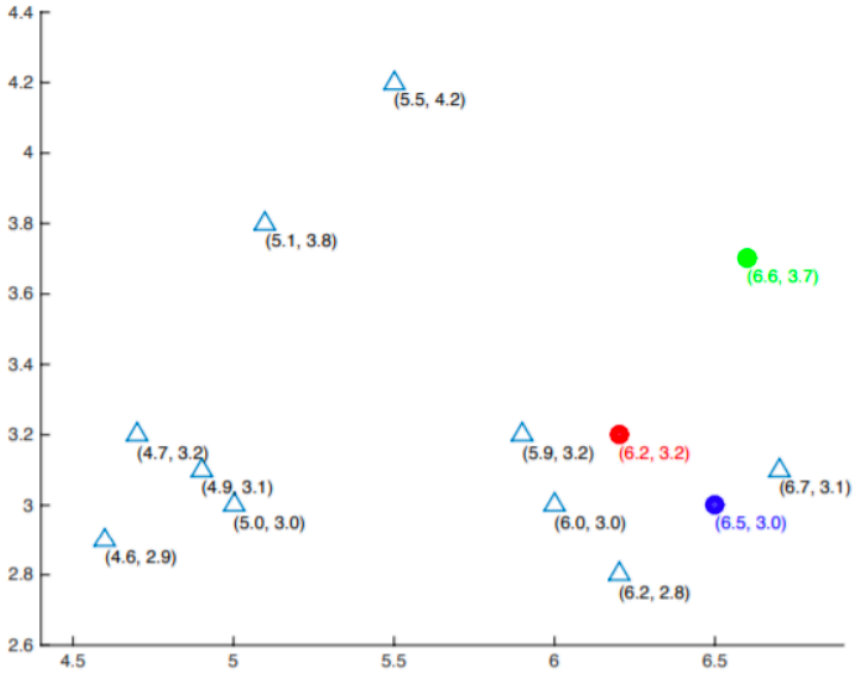


Figure 1: Scatter plot of datasets and the initialized centers of 3 clusters

Given the matrix X whose rows represent different data points, you are asked to perform a k-means clustering on this dataset using the Euclidean distance as the distance function. Here k is chosen as 3. The Euclidean distance d between a vector x and a vector y both in R^p is defined as $d = \sqrt{\sum_{i=1}^{p}(x_i - y_i)^2}$. All data in X were plotted in Figure 1. The centers of 3 clusters were

initialized as $\mu_1 = (6.2, 3.2)$ (red), $\mu_2 = (6.6, 3.7)$ (green), $\mu_3 = (6.5, 3.0)$ (blue).

$$X = \begin{bmatrix} 5.9 & 3.2 \\ 4.6 & 2.9 \\ 6.2 & 2.8 \\ 4.7 & 3.2 \\ 5.5 & 4.2 \\ 5.0 & 3.0 \\ 4.9 & 3.1 \\ 6.7 & 3.1 \\ 5.1 & 3.8 \\ 6.0 & 3.0 \end{bmatrix}$$

Solution :

(1) [10pt] What's the center of the first cluster (red) after one iteration? (Answer in the format

of [x_1,x_2], round results to three decimal places, same as part (2) and (3))

- ⇨ **Result after one iteration**
  Red Cluster

	X	**Y**
Mean	**5.171**	**3.171**

2) [5pt] What's the center of the second cluster (green) after two iteration?

- ⇨ **Result after two iterations**
  Green Cluster

	X	**Y**
Mean	**5.300**	**4.000**

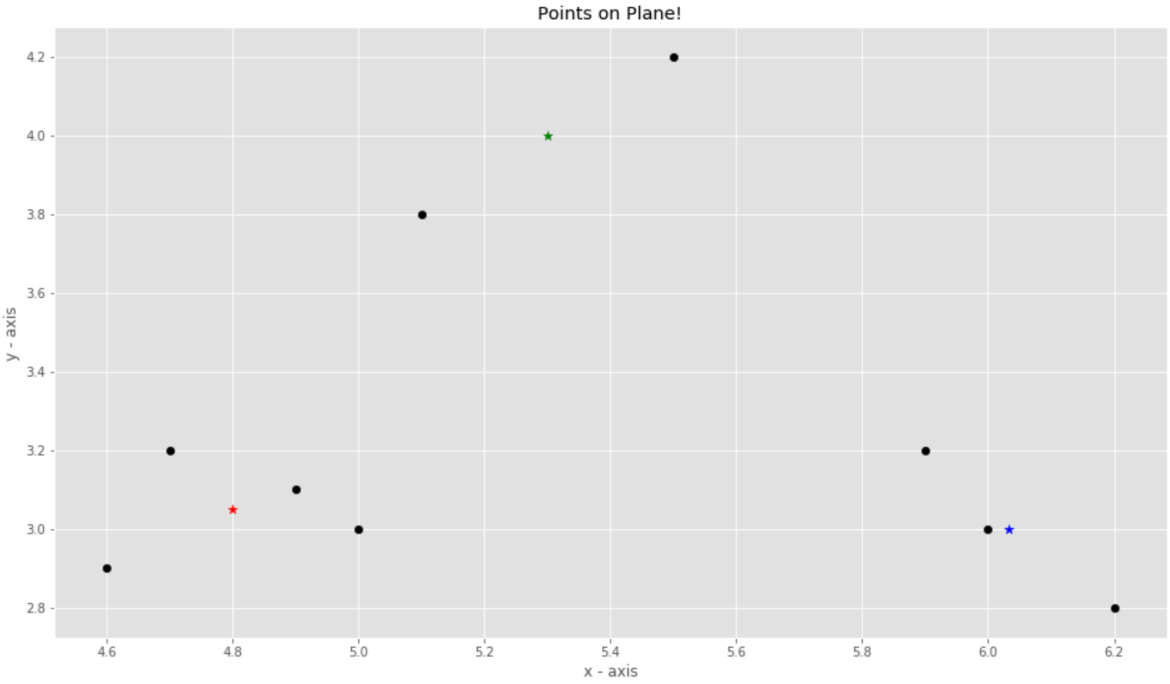
(3) [5pt] What's the center of the third cluster (blue) when the clustering converges?

⇨ **Result after clustering converge**
Blue Cluster

	X	Y
Mean	**6.033**	**3.000**

(4) [5pt] How many iterations are required for the clusters to converge?

⇨ **No. of iterations needed for clusters to converge is 2**



Problem 2 (15pt): [Latent variable model and GMM] Consider the discrete latent variable model where the latent variable **z** use 1-of-K representation. The distribution for latent variable **z** is defined as:

$$p(z_k = 1) = \pi_k$$

where $\{\pi_k\}$ satisfy: $0 \leq \pi_k \leq 1$ and $\sum_{k=1}^{K} \pi_k = 1$. Suppose the conditional distribution of observation **x** given particular value for **z** is Gaussian:

$$p(\mathbf{x}|z_k = 1) = \mathcal{N}(x|\mu_k, \Sigma_k)$$

(1) [5pt] Write down the compact form of $p(\mathbf{z})$ and $p(\mathbf{x}|\mathbf{z})$.

(2) [5pt] Show that the marginal distribution $p(\mathbf{x})$ has the following form:

$$p(\mathbf{x}) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}|\mu_k, \Sigma_k)$$

(3) [5pt] If we want to find the MLE solution for parameters π_k, μ_k, Σ_k in such model, what algorithm should we use? Briefly describe its major difference compared to K-means algorithm.

Solution :

1 . Compact form of p(z) and p(x|z).

Where the parameters {πk} must satisfy 0 <= πk <= 1 together with $\sum_{k=1}^{K} \boldsymbol{\pi_k} = \mathbf{1}$

Therefore ,

$p(z) = \prod_{k=1}^{K} \pi_k^{z_k}$ ------- (i)

$p(x|z) = \prod_{k=1}^{K} \mathrm{N}(x|\mu_k, \Sigma_k)^{z_k}$ ------- (ii)

2. Show the marginal distribution p(x) has the form :

$$p(\mathbf{x}) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}|\mu_k, \Sigma_k)$$

$p(x) = \sum_z p(z)p(x|z)$

Therefore from (i) and (ii),

$p(x) = \sum_{k=1}^{K} \boldsymbol{\pi_k}\, \boldsymbol{N}\,(x|\mu_k, \Sigma_k)$

3. MLE solution for parameters π_k, μ_k, Σ_k and difference to K-Means.

MLE for $p(x|\pi, \mu, \Sigma)$: ,

$\ln p(X|\pi, \mu, \Sigma) = \sum_{n=1}^{N} \ln \{\sum_{k=1}^{K} \boldsymbol{\pi_k}\, \boldsymbol{N}\,(x|\mu_k, \Sigma_k)\}$

Define a random variable $\gamma_k(X) = p(k|X)$,

$$= \frac{p(X|k)p(k)}{\sum_{k=1}^{K} p(k)p(X|k)}$$

$$= \frac{p(X|k)\boldsymbol{\pi_k}}{\sum_{k=1}^{K} \boldsymbol{\pi_k}\, p(X|k)}$$

Now for the log likelihood function to be maximum, its derivative of $p(X|\mu,\Sigma,\pi)$ with respect

to μ, Σ and π should be zero.
So equaling the derivative of $p(X|\mu, \Sigma, \pi)$ with respect to μ to zero and rearranging the terms,

$$\mu_k = \frac{\sum_{n=1}^{N} \gamma_k(x_n) x_n}{\sum_{n=1}^{N} \gamma_k(x_n)}$$

$$\Sigma_k = \frac{\sum_{n=1}^{N} \gamma_k(x_n)(x_n - \mu_k)(x_n - \mu_k)^T}{\sum_{n=1}^{N} \gamma_k(x_n)}$$

$$\pi_k = \frac{1}{N} \sum_{n=1}^{N} \gamma k(x_n)$$

Q .what algorithm should we use? Briefly describe its major difference compared to K-means algorithm.

Solution :

Here it is assumed that there are total N number of samples and each sample containing d features are denoted by x_i . So it can be clearly seen that the parameters cannot be estimated in closed form. This is where Expectation Maximization algorithm is beneficial. The Expectation-Maximization (EM) algorithm is an iterative way to find maximum-likelihood estimates for model parameters when the data is incomplete or has some missing data points or has some hidden variables. EM chooses some random values for the missing data points and estimates a new set of data. These new values are then recursively used to estimate a better first data, by filling up missing points, until the values get fixed . These are the two basic steps of the EM algorithm, namely E Step or Expectation Step or Estimation Step and M Step or Maximization Step. This is how it is different than K-Means Algorithm.

Problem 3 (20pt): [Bayesian Network] Suppose we are given 5 random variables, A_1, A_2, B_1, B_2, B_3. A_1 and A_2 are marginally independent and B_1, B_2, B_3 are marginally dependent on A_1, A_2 as follows: B_1 depends on A_1 and A_2, B_2 depends on A_2. B_3 depends on A_1. All 5 random variables are binary, i.e., $A_i, B_j \in \{0, 1\}, i = 1, 2; j = 1, 2, 3$.

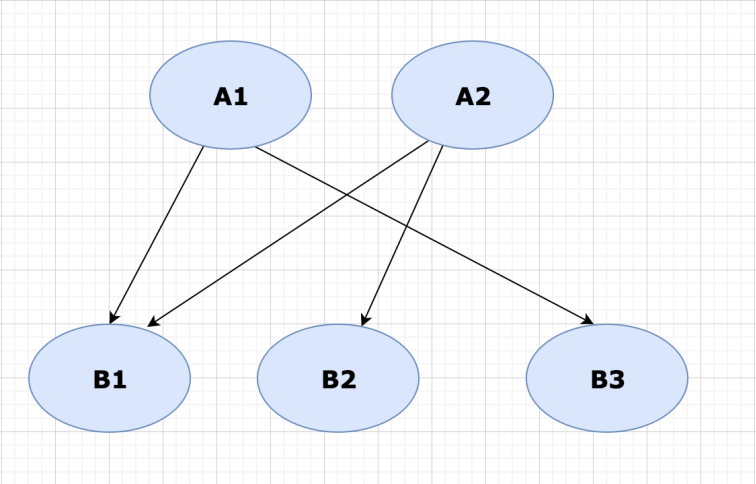
(1) [5pt] Draw the corresponding bayesian network.

(2) [5pt] Based on the bayesian network in (1), write down the joint distribution $p(A_1, A_2, B_1, B_2, B_3)$.

(3) [5pt] How many independent parameters are needed to fully specify the joint distribution in (2).

(4) [5pt] Suppose we do not have any independence assumption, write down one possible factorization of $p(A_1, A_2, B_1, B_2, B_3)$, and state how many independent parameters are required to describe joint distribution in this case.

1.Bayesian Network :



2. Joint distribution over any n variables requires $2^n - 1$ probability values.

$P(A_1, A_2, B_1, B_2, B_3)$

$= P(A_1)P(A_2 \mid A_1)P(B_1 \mid A_2 A_1)P(B_2 \mid B_1 A_2 A_1)P(B_3 \mid B_2 B_1 A_2 A_1)$

$= P(A_1)P(A_2)P(B_1|A_2A_1)P(B_2|A_2)P(B_3|A_1)$ 3. 1 + 1 + 4 + 2 + 2

= 10 independent parameters are needed

4. One such factorization is

$P(A_1)P(A_2|A_1)P(B_1|A_2A_1)P(B_2|B_1A_2A_1)P(B_3|B_2B_1A_2A_1)$ 1 + 2 + 4 + 8 + 16 = 31 independent parameters are needed

Problem 4 (40 pt) [Neural Networks]
Build a neural network with one hidden layer to predict class labels for Iris plant dataset (`https://archive.ics.uci.edu/ml/datasets/iris`).

(1) [30pt] Properly split the data to training and testing set, and report the training, testing accuracy. You can use sigmoid activation function and select any reasonable size of hidden units. Note that for this part, you need to implement the forward/backward function yourself without using the deep learning package. However, you can use the deep learning package, e.g., tensorflow, pytorch, matconvnet etc, to compare with your own results.

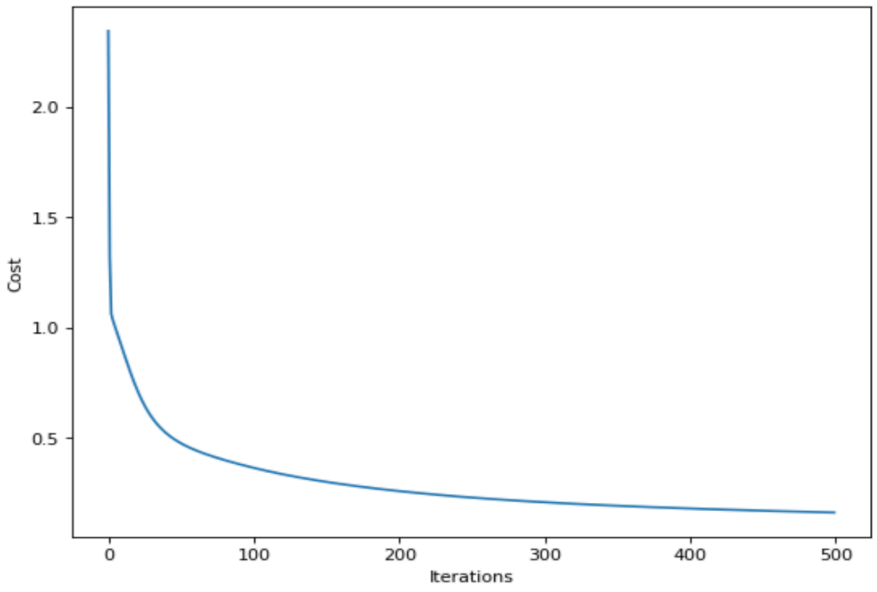
(2) [10pt] Try different design of the neural network, compare with part (1), and report findings. This is an open-ended question, you can change the previous model in several ways, e.g., (1) change the activation function to be tanh, ReLU etc, or (2) try to build more complex neural network by introducing more layers, or many other options. Note that for this part, you are allowed to use deep learning packages.

Solution :

Sigmoid :

```
Accuracy For Sigmoid:  90.0 %
F1 Score is  89.47368421052632 %
[[11  0  0]
 [ 0  8  2]
 [ 0  1  8]]
```

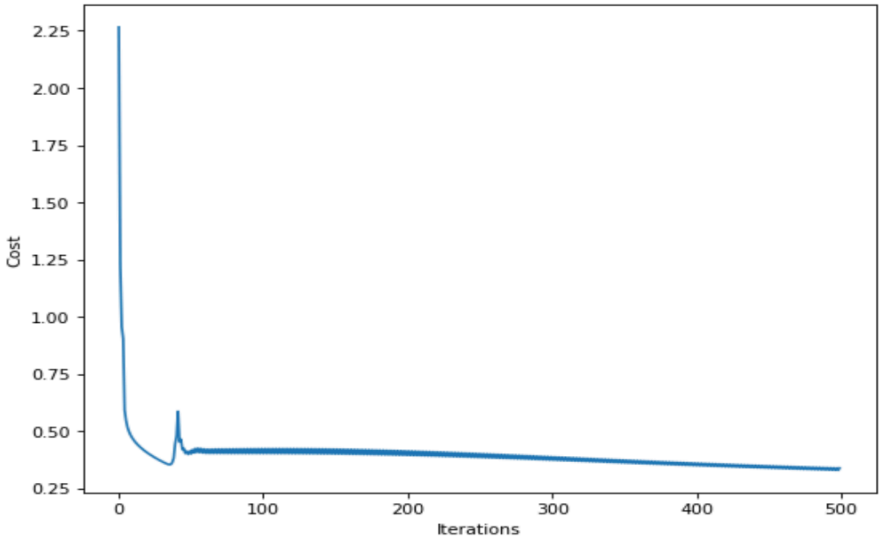
Iteration vs Cost Graph for ANN for multiclass Classification using Softmax and Sigmoid in hidden layer



Tanh :

```
Accuracy For Tanh:  86.66666666666667 %
F1 Score is  85.6060606060606 %
[[11  0  0]
 [ 0  9  4]
 [ 0  0  6]]
```

Iteration vs Cost Graph for ANN for multiclass Classification using Softmax and Tanh in hidden layer



Relu :

```
Accuracy For Relu:  83.33333333333334 %
F1 Score is  80.51282051282053 %
[[11  0  0]
 [ 0  4  0]
 [ 0  5 10]]
```

Iteration vs Cost Graph for ANN for multiclass Classification using Softmax and Relu in hidden layer