

Omar Abdelmotake

I pledge my honor that I have abided by the Stevens Honor System. *Omar A.*

CS559-B HW3

Due: Dec 7th, 2022

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

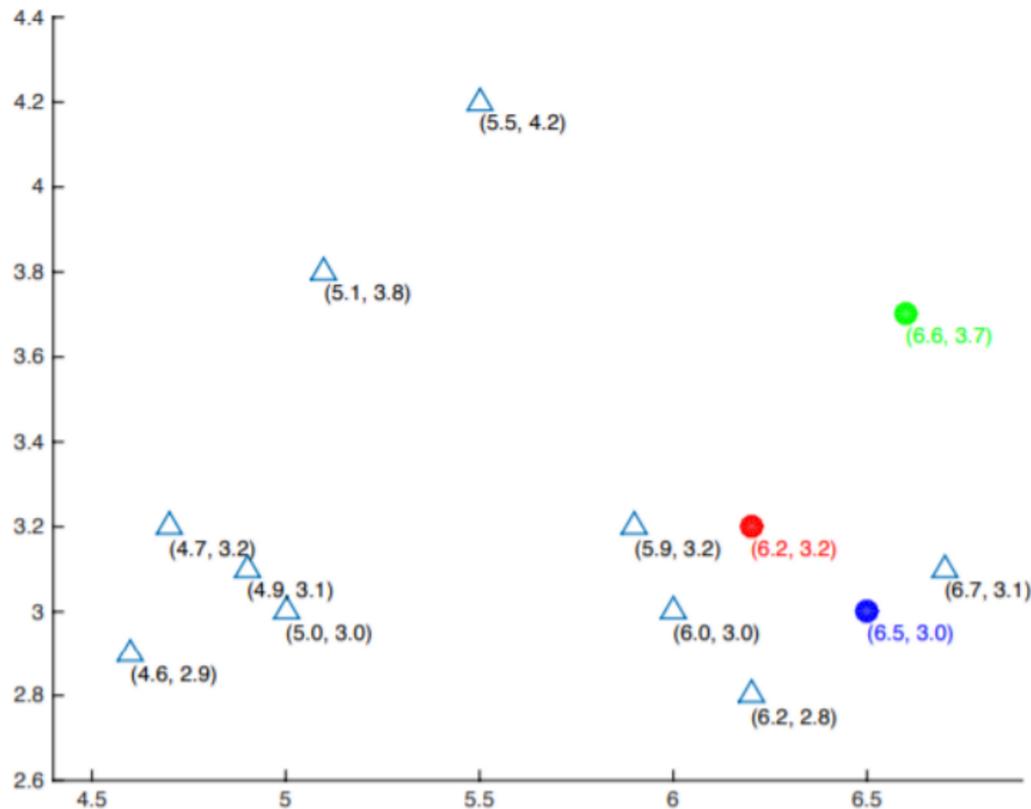


Figure 1: Scatter plot of datasets and the initialized centers of 3 clusters

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).

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

- (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))

$$(5.171, 3.171)$$

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

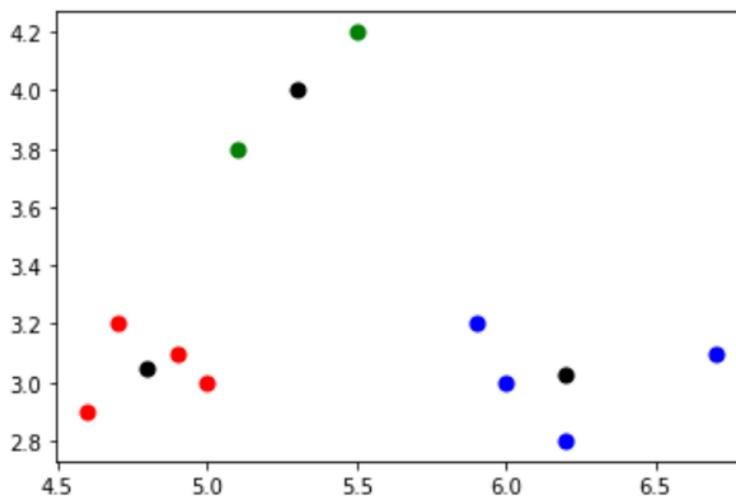
$$(5.300, 4.000)$$

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

(6.2, 3.025)

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

2 iterations required



Problem 2 (15pt): [Latent variable model and GMM] Consider the discrete latent variable model where the latent variable \mathbf{z} use 1-of-K representation. The distribution for latent variable \mathbf{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 \mathbf{x} given particular value for \mathbf{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 $p(\mathbf{x}|\mathbf{z})$.

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

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

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

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

$$p(\mathbf{x}) = \sum_{\mathbf{z}} p(\mathbf{z}) p(\mathbf{x}|\mathbf{z})$$

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

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

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

(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.

Expectation-maximization (EM) algorithm

$$\theta = \sum_{n=1}^N \frac{\pi_k N(x_n | \mu_k, \Sigma_k)}{\sum_j \pi_j N(x_n | \mu_j, \Sigma_j)} \underbrace{\sum_k (x_n - \mu_k)}_{r(z_{nk})}$$

$$N_k = \sum_{n=1}^N r(z_{nk})$$

$$\Rightarrow \mu_k = \frac{1}{N_k} \sum_{n=1}^N r(z_{nk}) x_n$$

$$\Rightarrow \Sigma_K = \frac{1}{N_n} \sum_{n=1}^N r(z_{nk})(x_n - \mu_k)(x_n - \mu_k)^T$$

$$\Rightarrow \pi_K = \frac{N_k}{N}$$

Compared to the k-means algorithm, the EM algorithm takes many more iterations and is less efficient.

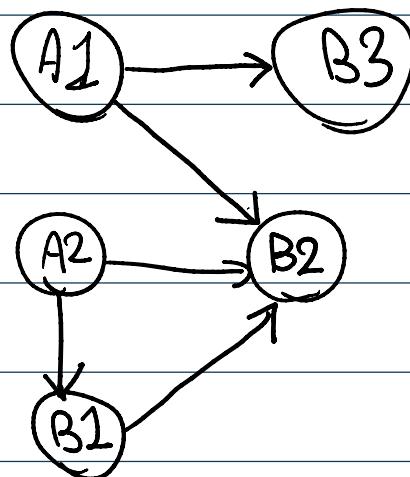
Problem 3 (20pt): [Bayesian Network] Suppose we are given 5 random variables, A_1, A_2, B_1, B_2, B_3 . A_1 and A_2 are marginally independent and B_1, B_2, B_3 are marginally dependent on A_1, A_2 as follows: B_1 depends on A_2 , B_2 depends on A_2, A_1 and B_3 depends on A_1 . All 5 random variables 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.

B_1 depends on A_2 .

B_2 depends on A_2, A_1, B_1 .

B_3 depends on A_1 .



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

$$p(A_1, A_2, B_1, B_2, B_3) =$$

$$p(A_1)p(A_2)p(B_1 | A_2)p(B_2 | A_1, A_2, B_1)p(B_3 | A_1)$$

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

Since there can only be two values $\{0, 1\}$ and there are 5 random variables, there would be $2^5 = 32$ possible independent parameters.

(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.

With no independence assumption, each probability can be written individually.

$$p(A_1, A_2, B_1, B_2, B_3) = p(A_1)p(A_2)p(B_1)p(B_2)p(B_3)$$

They are all binary, so there are 5 independent

They are all binary, so there are 5 independent parameters.

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.

(1) Training accuracy: 0.908

Testing accuracy: 0.700

(2) The loss averaged to about 50% each run with 5 epochs using tensorflow.