CS559-B HW3

Due: Dec 9th, 2020

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

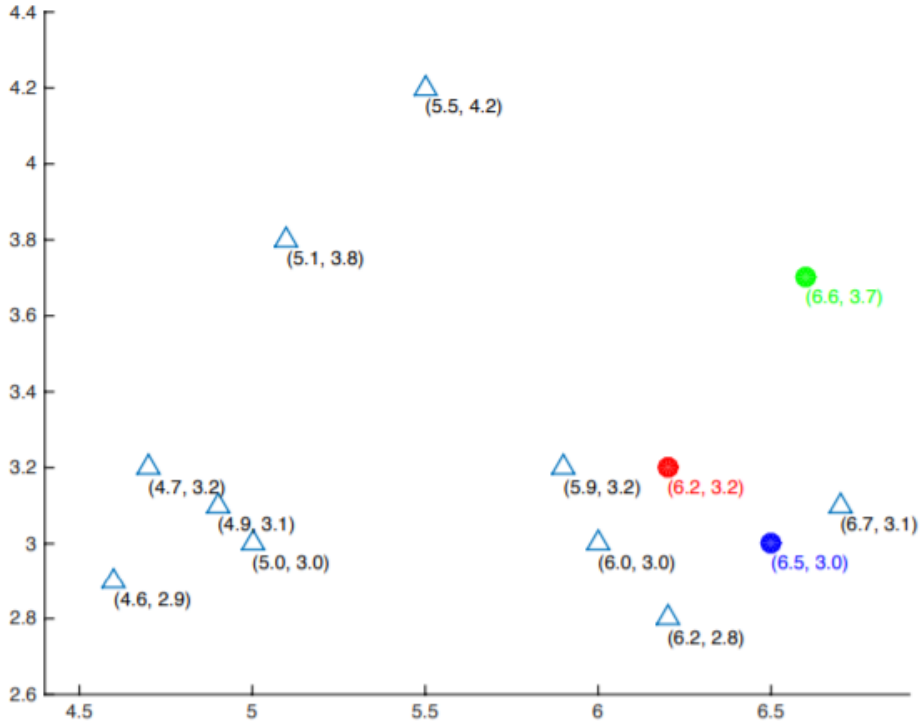

Figure 1: Scatter plot of datasets and the initialized centers of 3 clusters

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))

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

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

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

Problem 2 (15pt): [K-means and gradient descent] Recall the loss function for k-means clustering with k clusters, sample points x_1, x_2,..., x_n, and centers μ_1, μ_2,..., μ_k:

$$L = \sum_{j=1}^{k} \sum_{x_i \in S_j} ||x_i - \mu_j||^2$$

, where S_j refers to the set of data points that are closer to μ_j than to any other cluster mean.

(1) [5pt] Instead of updating μ_j by computing he mean, let's minimize L with *batch* gradient descent while holding the sets S_j fixed. Derive the update formula for μ_1 with learning rate ϵ.

(2) [2pt] Derive the update formula for μ_1 with *stochastic* gradient descent on a single sample point x_i. Use learning rate ϵ.

(3) [8pt] In this part, we will connect the batch gradient descent update equation with the standard k-means algorithm. Recall that in the update step of the standard algorithm, we assign each cluster center to be the mean of the data points closest to that center. It turns out that a particular choice of the learning rate ϵ (which may be different for each cluster) makes the two algorithms (batch gradient descent and the standard k-means algorithm) have identical update steps. Let's focus on the update for the first cluster, with center μ_1. Calculate the value of ϵ so that both algorithms perform the same update for μ_1.

Problem 3 (10pt): [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) [2pt] Write down the compact form of $p(\mathbf{z})$ and $p(\mathbf{x}|\mathbf{z})$.

(2) [3pt] 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.

Problem 4 (20pt): [Bayesian Network] Suppose we are given 5 random variables, A_1, A_2, B_1, B_2, B_3. These random variables have dependence relations as follows: B_1 depends on A_1, A_2 depends on B_1. B_2 depends on A_2 and A_1. B_3 depends on A_2. 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.

Problem 5 (30 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) [20pt] 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.