Lab 8: K-Means Clustering

Overview of algorithm

1. Randomly choose K centroids (in practice we often choose them from the existing data points)
2. Calculate the distance of all instances to the K centroids and assign instances to closest centroid
3. Calculate new centroid for each of the K clusters
4. Repeat Step 2 and 3 until clusters assignments are stable or centroids are not changing

The MNIST dataset is a dataset of 28 × 28 images (784 pixels) of hand-written digits. To read these images in Python, you can use the following script:

```
from sklearn.datasets import fetch_openml
    mnist = fetch_openml("mnist_784", version=1)
X = mnist["data"]
```

Since the dataset is quite large, restrict yourself to the first 2,000 training images. (2000 x 784)

Requirements

a. Write a function my_kmeans to perform a k-means clustering of the 2000 images of digits.

b. Your function should take 3 arguments, the data matrix, the number of clusters K, and the number of initializations M.

   (1) Your code should consist of 3 nested loops.

   (2) The outermost (from 1 to M) cycles over random centroids initializations (i.e. you will call k-means M times with different initializations).

   (3) The second loop is the actual k-means algorithm for that initialization, and cycles over the iterations of k-means.

   (4) Inside this are the actual iterations of k-means. Each iteration can have 2 successive loops from 1 to K: the first assigns observations to each cluster and the second recalculates the means of each cluster.

c. Your function should return:

   (1) the K centroids and cluster assignments for the best solution with the lowest loss function (recall that the k-means loss function is the sum of the squared distances of observations from their assigned means)

   (2) the sequence of values of the loss-function over k-means iterations for the best solution (this should be non-increasing)

   (3) the set of M terminal loss-function values for all initializations

d. Run your code on the 2000 digits for K = 10 and M = 15. Plot the sequence of values of the loss-function over k-means iterations for the best solution.

e. Plot the M terminal loss-function values for all initializations.