K-means Clustering

1 Theory

K-means is a clustering algorithm. Clustering algorithms are unsupervised techniques for sub-dividing a larger data set into smaller groups.

The k-mean clustering includes some steps:

- Step 1: Initlialize randomly centroids of clusters using points from the dataset.
- Step 2: Assign datapoints to the clusters using a distrance metric.
- Step 3: Compute new centroids using the mean of the clusters.
- Repeat step 2 and step 3 until there is no change in the clusters.

For a step-by-step visualization of K-means, we recommend this online tool: `http://stanford.edu/class/ee103/visualizations/kmeans/kmeans.html`

2 Python Exercise

In this exercise, you will implement k-means clustering algorithm to subdivide the `sklearn`'s dataset *digits*. The dataset contains images of handwritte digits from 0 to 9. Table 2 provides information regarding this dataset.

You are going to do the following steps.

- Step 1: In this step, you will load the dataset *Digits* with 5 classes. You can choose how much data you want to use, but the whole dataset will take you more time to process. Then, you will visualize some images in this dataset using `matplotlib`. Also, you will be provided the

Table 1: *Digits* dataset.

Classes	10
Samples per class	~ 180
Samples total	1797
Dimensionality	64
Features	integers $0 - 16$

code to visualize the dataset using `t-SNE`, which is one of the strongest method for visualizing high dimensional data. An optional step is to try different data reduction methods for visualization such as Principal Component Analysis (PCA).

- Step 2: You have to implement function `kmeans` in this step. The function takes two inputs: the data from `Digits` and the number of clusters. The expected outputs are the centroids of clusters and the labels of the data.

- Step 3: This step will evaluate your clustering results. You will use the function `kmeans` to make clusters. Also, you will visualize some images from the clusters to make sure that members of a cluster are indeed similar.

- Step 4: In this step, you will calculate the *silhouette* score. *Silhouette* is a metric to measure the quality of clustering. The score is bounded between -1 for incorrect clustering and +1 for highly dense clustering. Scores around zero indicate overlapping clusters. The score is higher when clusters are dense and well separated, which relates to a standard concept of a cluster. You can use the implemetation from `sklearn` for this score.