COSC74/174: Machine Learning and Statistical Data Analysis
Homework 4. Due: 11:59PM on Friday, 15 Nov 2019
Instructor: Prof. Soroush Vosoughi

Unsupervised Learning

You are given a training dataset in CSV format (hw4_data1.csv). The files each contain 40 rows with 2 columns. Column 1 & 2 are the features. There are no labels for this dataset.

Your goal for this assignment is to implement different clustering algorithms and run them on this dataset. For this assignment you can assume the distance function is the Manhattan distance.

Part a) [30 points] Implement K-Means/Median. Implement a generalized K-means/median algorithm. You should have a single function that takes in as input the data points, K, and some other hyperparameters, specified below. The function should return K lists of data points. Each list corresponds to one cluster.

The hyperparameters your functions should support and the values they can take are:

The method for calculating the centroid: Means or Median

The initialization method: Random Split Initialization or Random Seed Selection Method

Max_iterations: max number of iterations to run the algorithm.

K: number of clusters

```
def  K_means_family (datapoints, K, centroid_method, initialization method, max_iterations):

                return list of clusters
```

Note that your stopping condition should have two parts: 1) stop if you pass the max iterations 2) stop if no change is made in the last step.

You will be running this code in part c of the assignment. For this part you just need to implement the function.

Part b) [10 points] Silhouette score

In this part of the assignment you are implementing a function that calculates the Silhouette score for a list of clusters. The function should take in a list of clusters (such as the output of the last function you implemented) and return a single Silhouette score.

```
def Silhouette(list_of_clusters):

        return Silhouette_score
```

Part c) [10 points] Finding Best K

Run the code you implemented in part a for k=2,3,4,5. Set the other hyperparameters to the following:

The method for calculating the centroid: Means

The initialization method: Random Split Initialization

Max_ iterations: 100

Calculate the silhouette score for each K using the function in part b and use these scores to pick the best K. What is the best K?

Part D) BONUS [25 points] Hierarchical Agglomerative Clustering

Implement the hierarchical clustering algorithm covered in the class. The algorithm takes one hyperparameter as input: function to calculate distance between clusters: MIN, MAX, AVG (look at the slides to be reminded of what these mean).

Your function should run until there is only one cluster that contains all the datapoints. The function should return a list containing the list of clusters at each step. E.g., if you have two datapoints as your input, the function should return:

[[[d1],[d2]] , [[d1,d2]]]

The function should also return a list containing the dimeter of the cluster that was created at each merge. So for the dataset above, it will be a list with a single element, which is the diameter of the [d1, d2] cluster.

```
def HAC (datapoints, distance_function):

        return cluster_history, diameters
```

Run this algorithm on the given dataset with the distance function set to AVG . Save the output of the algorithm to two files, one containing the clusters and the other the diameters. The file containing the clusters should be in order (i.e., the first line should be the set of singleton clusters and the last line should be one cluster containing all points). The diameters should also be written in order.