Assignment 3: Clustering

Due: November 10 2023 11:59PM

Four datasets (Iris, YeastGene, Example, Utilities) are provided. In each dataset, each row corresponds to an object and each column corresponds to an attribute. The attribute values are comma-separated. You don't need to do normalization for any of the datasets.

The initial centroids to be used in K-means for Iris and YeastGene datasets are provided. In each file, each row denotes the initial centroid of a cluster.

In this assignment, you are asked to implement K-means algorithm and Agglomerative algorithm (using Min to define inter-cluster distance). Templates (*kmeans_template.py, hierarchical_template.py*) are for Python 3.

In *kmeans_template.py*, you are asked to fill in two functions: *assignCluster* and *getCentroid*. In *assignCluster*, each object is assigned to the cluster whose centroid is the closest to the object, based on Euclidean distance. In *getCentroid*, cluster centroids are updated based on current assignment.

In *hierarchical_template.py*, you are asked to fill in *merge_cluster* and *update_distance*. In *merge_cluster*, you need to merge two closest clusters. In *update_distance*, you need to update the distance matrix after the merging using Min as the inter-cluster distance.

If you are not sure about whether it is OK to use a certain function, please post your question on Piazza.

Please take the following steps:

1. Implement K-means algorithm as follows:

```
 Repeat T times:
   • For each object $x_i$
        ▪ Calculate Euclidean distance between $x_i$ and each of the $K$ centroids
        ▪ Assign $x_i$ to the cluster whose centroid is the closest to $x_i$
   • For each cluster
        ▪ Calculate its centroid as the mean of all the objects in that cluster
```

2. Implement Hierarchical clustering algorithm (with Min as inter-cluster distance definition):

- Obtain the distance matrix by computing Euclidean distance between each pair of objects
- Let each object be a cluster (Assign the cluster index as 1 to N, where N is the number of objects)
- Set the current index as T=N+1

- **Repeat**
    - Find the smallest value in the distance matrix—suppose the entry is *i*-th row and *j*-th column, so the *i*-th and *j*-th data points are the closest at the current step.
    - Find the clusters that contain the *i*-th data point (the *s*-th cluster) and the *j*-th data point (the *t*-th cluster).
    - Merge the *s*-th and *t*-th clusters as a new cluster with index *T* and remove the *s*-th and *t*-th clusters.
    - For each pair of data points between the the *s*-th and *t*-th clusters (one point in the *s*-th cluster, and one point in the *t*-th cluster), change their distance to a big number.
    - *T*=*T*+1
- **Until only one cluster remains**

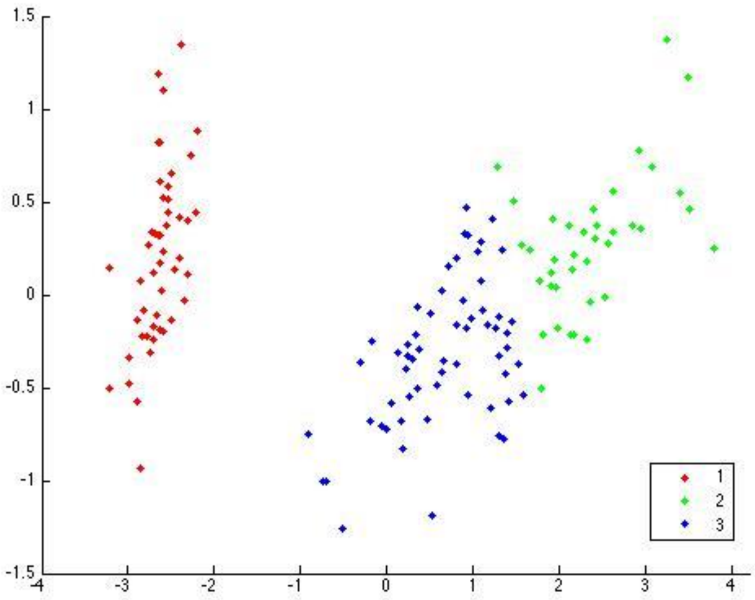
3. Test your K-means on Iris dataset. Use the provided initial centroids, and set the number of iterations (*T*) as 12, and the number of clusters (*K*) as 3. The final cluster centroids should be:

Cluster 1: 5.006,3.418,1.464,0.244

Cluster 2: 6.8538,3.0769,5.7154,2.0538

Cluster 3: 5.8836,2.741,4.3885,1.4344

4. After running the filled *kmeans_template.py* code file, you can obtain a data file *Iris_ kmeans_cluster.csv*, which contains the Iris data and the assigned clusters. Then, apply the PCA and plotting functions you used in Assignment 1 to project Iris data to 2 dimensions and draw the scatter plot. Use different colors for different clusters in the scatter plot. The plot should look like the one below:



5. If you get the correct final cluster centroids and the plot, then repeat steps 3 and 4 on the YeastGene dataset. Use the provided initial centroids, and set the number of iterations (*T*) as 7, and the number of clusters (*K*) as 6.

6. Apply Hierarchical Clustering algorithm implemented in Step 2 on the Example dataset. The order of the merging should be:

1	2	7
3	7	8
4	5	9
6	9	10
8	10	11

Each row here denotes an operation of merging two clusters to form a new cluster. The first two indices denote the clusters to be merged, and the last one denotes the index of the new cluster.

7. If you get the correct order in Step 6, then apply the hierarchical clustering algorithm on the Utilities dataset.

8. Prepare your submission. Your final submission should be a zip file named as Assignment3.zip. In the zip file, you should include:

- The Python code.
- Report: A WORD or PDF file named as Assignment3.docx or Assignment3.pdf. The report should consist of the following parts: 1) The cluster centroids obtained on YeastGene dataset after the *T* iterations. 2) The scatter plot obtained on YeastGene dataset after applying PCA and plotting points using different colors for different clusters. 3) The order of merging in the hierarchical clustering on the Utilities dataset. 4) The codes of your K-means and hierarchical clustering algorithm implementation (i.e., the four functions: *assignCluster*, *getCentroid*, *merge_cluster* and *update_distance*).

9. Submit the zip file under Assignment 3 on Brightspace.

Please refer to Course Syllabus for late submission policy and academic integrity policy. This assignment must be done independently. Running your submitted code should be able to reproduce the results in the report.