CAP4770 Intro To Data Science
Assignment 5

Description

This assignment aims to implement some clustering algorithms using the Julia language and analyzing their performance(score) and the results. We will use the following algorithm for the assignment:

1. K Means
2. Kmedoids
3. Hierarchical clustering
4. fuzzy_Cmeans
5. Gaussian Mixture Models

Libraries required

DataFrames

ScikitLearn

LinearAlgebra

Clustering

CSV

RDatasets

Distances

Statistics

GaussianMixtures

Running jupyter notebook

1. Unzip the file Assignment5.zip
2. Then run “cd Assignment5” in the terminal.
3. Run “jupyter notebook” in the terminal.
4. Open file “Clusters.ipynb”.
5. Now, run each cell from top to bottom one by one so that there will be no errors. Do not skip any cell which can cause errors.

Dataset

The data consists of 22 numerical and 5 categorical attributes. First, we will convert all the categorical data into numerical data and then will normalize each and every column on the scale of 0.0 to 1.0.

Now, the dataset is processed and we will implement all algorithms.

For scoring, we have used silhouettes. **Silhouette** is a method for evaluating the quality of clustering. Particularly, it provides a quantitative way to measure how well each point lies within its cluster in comparison to the other clusters.

All the recordings below can differ with each execution as everything depends on choosing starting cluster centers which are random.

K Means Clustering

1. For k means, we have used 3 clusters so as to get the best score. Maximum iterations are set to 200 and we will transpose the data required for the data processing for k means. The final, function will look like

   kmeans(data',3, maxiter=200)

2. The score for kmeans is **0.2888818238308511**
3. Running time for kmeans is **0.045236 seconds**
4. We have created 3 clusters with the following sizes:
   a. 1684
   b. 5136
   c. 1705

After computing the centers for each feature, clusters seem to be not very far apart from each other.

5. 3 outliers points are displayed on the jupyter notebook. Outliers are detected using the silhouette score. As provides a quantitative way to measure how well each point lies within its cluster in comparison to the other clusters. The point with **negative** silhouettes score is outliers.

Kmedoids Clustering

1. For kmedoids, we have used 3 clusters so as to get the best score. Maximum iterations are set to 200 and we converted the original matrix

to distance matrix by calculating the squared euclidean distance for each point to every other point.

kmedoids(distances,3, maxiter=200)

2. The score for kmedoids is **0.056080511756002344**
3. Running time for kmedoids is **8.257388 second**
4. We have created 3 clusterings with the following sizes:
    a. 2693
    b. 2990
    c. 2842

    The index of 3 medoids for each cluster are

    4628, 5074,4097.

5. 3 outliers points are displayed on the jupyter notebook. Outliers are detected using the silhouette score. As provides a quantitative way to measure how well each point lies within its cluster in comparison to the other clusters. The point with **negative** silhouettes score is outliers.

Hierarchical Clustering

1. For Hierarchical Clustering, we have used a single linkage method to build the dendrogram.

    hclust(distances, linkage=:single)

2. The score for this algorithm is **0.5540756582058237**
3. The running time for this algorithm is **8.720685 seconds**
4. We have created 2 clusters. Not much can be interpreted from the results but the dendrogram values show that most of the points in the cluster are leaf nodes with **negative merges value(explained in jupyter notebook as a comment).**
5. 3 outliers points are displayed on the jupyter notebook. Outliers are detected using the silhouette score. As provides a quantitative way to measure how well each point lies within its cluster in comparison to the other clusters. The point with **negative** silhouettes score is outliers.

Fuzzy_Cmeans Clustering

1. For fuzzy_cmeans, 2 clusters for best score

    fuzzy_cmeans(distances, 2, 2, maxiter=200, display=:iter)

2. The score for fuzzy is **0.20557593481435799**

3. Running time for fuzzy is **9.616789 seconds**
4. After computing the centers for each feature, clusters seem to be not very far apart from each other.
5. 3 outliers points are displayed on the jupyter notebook. Outliers are detected using the silhouette score. As provides a quantitative way to measure how well each point lies within its cluster in comparison to the other clusters. The point with **negative** silhouettes score is outliers.

Gaussian Mixture Models

For initializing a GMM constructor, we have used a splitting method with 4 components as a Gaussian mixture

The score for GMM is **4.242033971215994.** We have used average log-likelihood for the scoring of gmm. In statistics, the likelihood function measures the goodness of fit of a statistical model to a sample of data for given values of the unknown parameters.

Running time for the GMM is **0.564757 seconds**