Birla Institute of Technology and Science-Pilani,Hyderabad Campus

Second Semester 2017-18

Data Mining (CS F415)

Hierarchical Clustering -Agglomerative and Divisive
Partitional Clustering-K medoids

By

CHINTA NIKHIL 2016A7PS0026H
ANISH REDDY 2016A7PS0104H
Mahaditya Kaushik 2016B1A70723H

Data Set Used:

Human gene DNA Sequence

Data pre-processing:

File was read and the DNA sequences were stored in a dictionary, where the key is the gene sequence's name and the value contains the entire gene string.A mapping was created from the unique gene sequences in the dataset to integers so that each sequence corresponded to a unique integer.

Distance Matrix:

Distance Matrix is an NxN matrix where a point (i, j) denotes the alignment distance between the i th and the j th DNA sequence strings.Due to slower computation power of Python over C++, computation of edit distance between DNA sequence strings takes a much longer time (around 22000 seconds). As a result, this computation can not be repeated. Hence the distance matrix is stored as a pickle file to be reused later

Linkage Matrix:

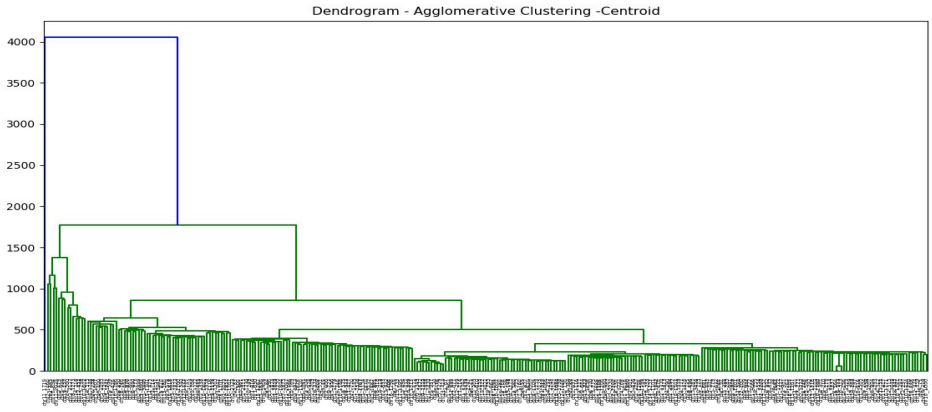
Scipy uses a special matrix called linkage matrix to draw dendrograms. The shape of the matrix is 2N-2 x 4, where the i th row represents the merging of two clusters to form the (n+i) th cluster. The first and second columns of the matrix contain the clusters being merged, the third column contains the distance between the two clusters being merged, and the fourth column contains the number of elements in the merged cluster.

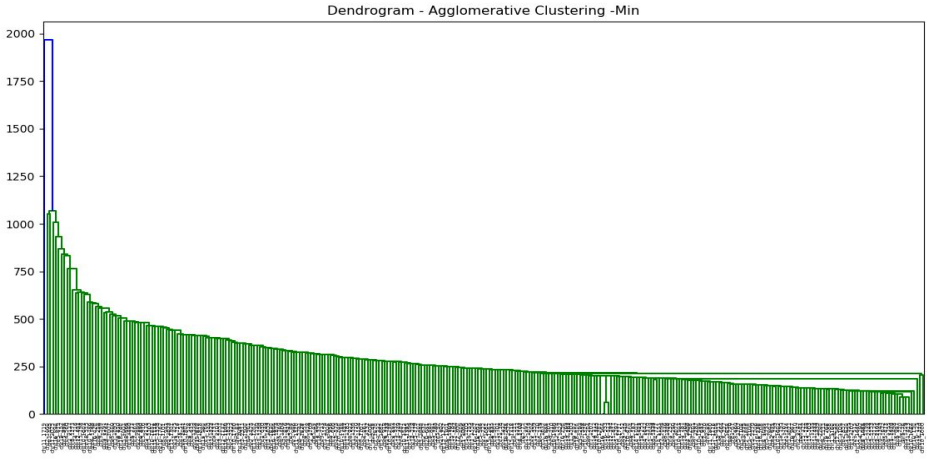
Agglomerative Clustering:

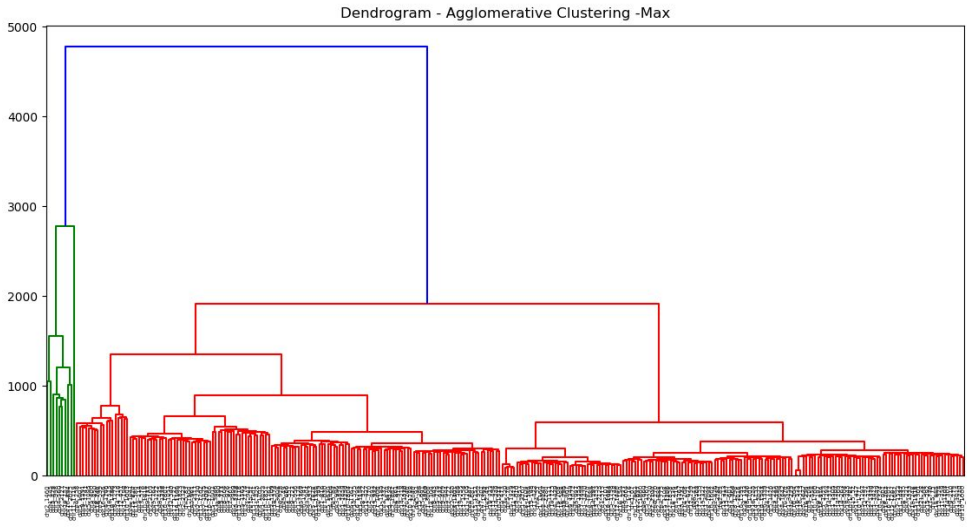
Formulas Used:

Maximum or complete-linkage clustering - Max(d(a,b))
Minimum or single-linkage clustering - Min(d(a,b))
Mean or average linkage clustering - sum of all d(a,b)/|A|+|B|

RESULTS:





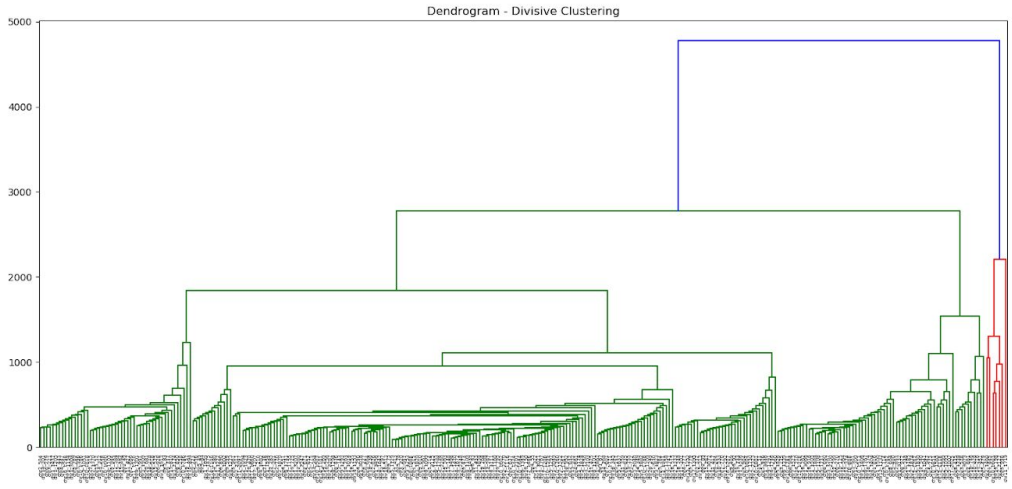


Divisive Clustering:

Formulas Used:

Diameter of a cluster - Max(d(a,b))

RESULTS:



Comparison of Agglomerative and Divisive Clustering:

Bottom-up(Agglomerative) clustering is much faster as compared to top down clustering.

• Agglomerative clustering completes execution in polynomial time
• Divisive clustering requires exponential time
•Agglomerative Clustering cannot undo what has been done previously. If two clusters have been combined, they cannot be separated again.
The dendrograms generated by top-down and bottom up clustering are not same, but they are similar.
•In top-down hierarchical clustering(Divisive), we divide the data into 2 clusters (using k-means with k=2k=2, for example). Then, for each cluster, we can repeat this process, until all the clusters are too small or too similar for further clustering to make sense, or until we reach a preset number of clusters.

•In bottom-up hierarchical clustering(Agglomerative), we start with each data item having its own cluster. We then look for the two items that are most similar, and combine them in a larger cluster. We keep repeating until all the clusters we have left are too dissimilar to be gathered together, or until we reach a preset number of clusters.

Comparison of Hierarchical Clustering and K-Medoids

• k means(Medoids) clustering is the process of partitioning a set of data objects into subsets. Each subset is a cluster such that the similarity within the cluster is greater and the similarity between the clusters is less.
• In Partitioning methods, we find the mutually exclusive cluster of spherical shape based on distance. In this case, we can use mean or median as a cluster centre to represent each cluster. It is helpful in the small and medium size of data.

•In Hierarchical methods, we create hierarchical decomposition of the given set of data. We create hierarchical decomposition in two ways such as from bottom to the top or top to down. On the basis how we create hierarchical decomposition we divide this method into two approaches one is agglomerative approach and other is the divisive approach.

• The main problem of this process is once a step is done it can never be undone.

•In Density-Based methods, we just find arbitrarily shaped clusters which are dense regions of objects in space that are separated by low-density regions.

•In this method, each point must be belonging in the strongly dense neighbourhood which detects outlier so easily. Grid-based methods quantize the object space into a finite number of cells that form a grid structure without depending on the number of the data objects

LANGUAGE USED : PYTHON

PYTHON LIBRARIES USED :
numpy,pyplot,pathlib,Dendogram,math,argparse,mathplotlib,pickle.

NOTE:

We have used levenshtein distance(Edit Distance) to calculate the distance between two strings.

ALGORITHM USED for Agglomerative:

```
SIMPLEHAC(d_1, ..., d_N)
 1  for n ← 1 to N
 2  do for i ← 1 to N
 3      do C[n][i] ← SIM(d_n, d_i)
 4      I[n] ← 1  (keeps track of active clusters)
 5  A ← []  (assembles clustering as a sequence of merges)
 6  for k ← 1 to N − 1
 7  do ⟨i, m⟩ ← arg max_{⟨i,m⟩: i≠m ∧ I[i]=1 ∧ I[m]=1} C[i][m]
 8      A.APPEND(⟨i, m⟩)  (store merge)
 9      for j ← 1 to N
10      do C[i][j] ← SIM(i, m, j)
11          C[j][i] ← SIM(i, m, j)
12      I[m] ← 0  (deactivate cluster)
13  return A
```

ALGORITHM USED for Divisive: