

Birla Institute of Technology and Science-Pilani, Hyderabad
Campus

Second Semester 2017-18



Data Mining (CS F415)

Hierarchical Clustering – Agglomerative and Divisive
Clustering – K Means

by

Vrutik Halani 2017A7PS1732H Shreyansh

Garg 2017A7PS1730H Kunal Verma

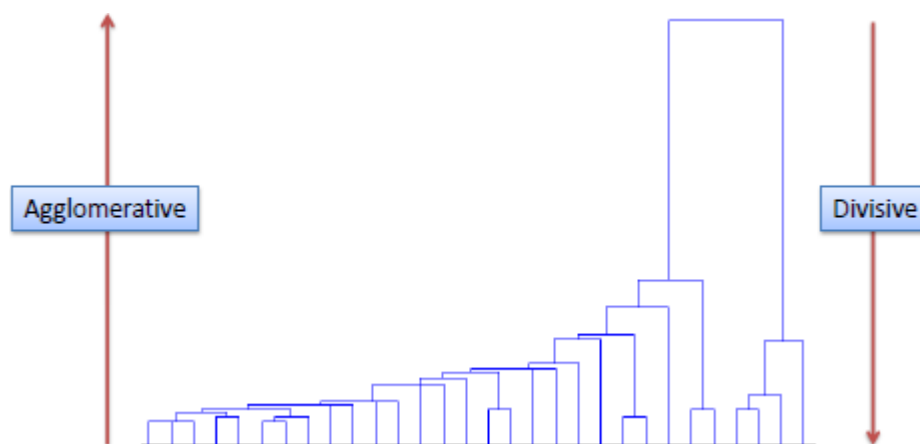
2017A7PS0120H Sandesh Thakar

2017A7PS0181H

CLUSTERING

Cluster analysis or clustering is the task of grouping a set of objects in such a way that objects in the same cluster are more similar to each other than to those in other clusters.

Hierarchical clustering involves creating clusters that have a predetermined ordering from top to bottom. For example, all files and folders on the hard disk are organized in a hierarchy. There are two types of hierarchical clustering: *Divisive* and *Agglomerative*.



DIVISIVE METHOD

In *divisive* or *top-down clustering* method we assign all of the points to a single cluster and then partition the cluster to two least similar clusters. Finally, we proceed recursively on each cluster until there is one cluster for each point.

AGGLOMERATIVE METHOD

In *agglomerative* or *bottom-up clustering* method, we start with individual points as a cluster. Then, compute the distance between each of the clusters and join the two most similar clusters until we are only left with single cluster consisting of all the points.

K-Means

K-means clustering is a type of unsupervised learning, which is used when you have unlabeled data (i.e., data without defined categories or groups). The goal of this algorithm is to find groups in the data, with the number of groups represented by the variable K. The algorithm works iteratively to assign each data point to one of K-groups based on the features that are provided. Data points are clustered based on feature similarity.

Data Set Used

Amino Acid Sequence

Distance Matrix

Distance Matrix is an $N \times N$ matrix where a point (i, j) denotes the alignment distance between the i^{th} and the j^{th} Amino acids sequence strings.

Linkage Matrix

Scipy uses a special matrix called linkage matrix to draw dendrograms. The shape of the matrix is $2N-2 \times 4$, where the i^{th} row represents the merging of two clusters to form the $(n+i)^{\text{th}}$ 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.

I. Agglomerative Clustering

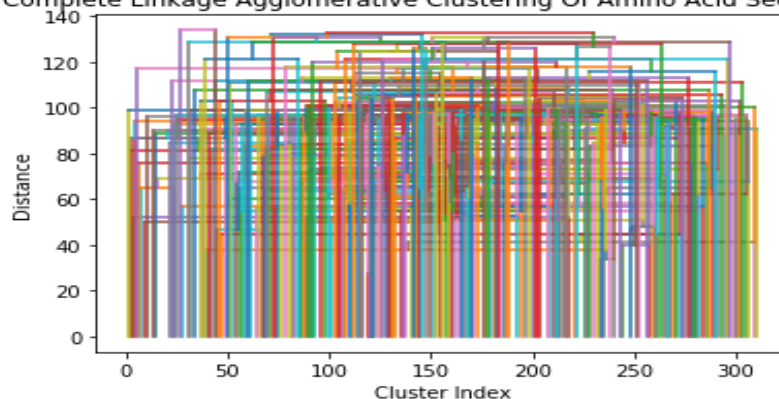
Formulas Used

Rows in the distance matrix are merged when we form a cluster and points in the newly formed cluster are modified so that they are not taken up for consideration again while forming new cluster. The newly formed cluster will have distances to all points outside the cluster using one of the heuristic (min, max, mean).

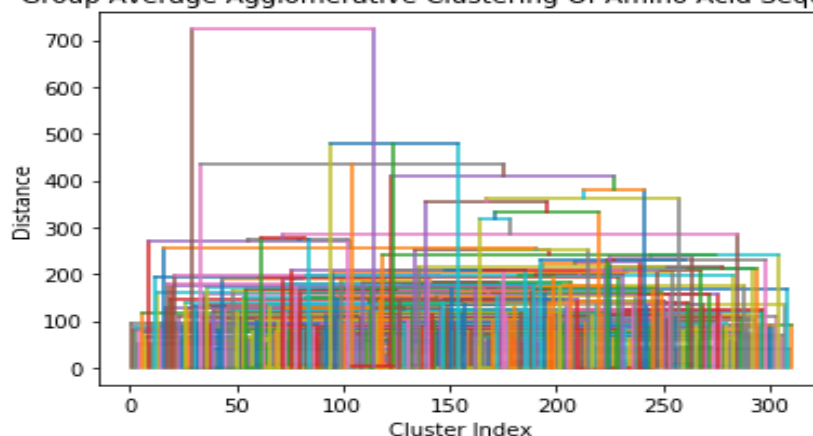
MODIFICATION AFTER CLUSTERING	FORMULA
Maximum or complete-linkage clustering	$\text{Max}(d(a,b))$
Minimum or single-linkage clustering	$\text{Min}(d(a,b))$
Mean or average linkage clustering	$\frac{\text{sum of all } d(a,b)}{ A + B }$

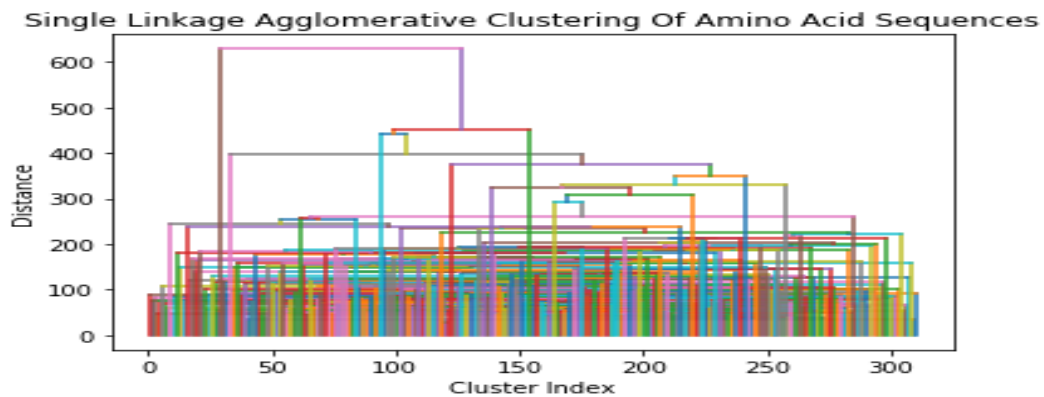
**a belongs to A, b belongs to B*

Complete Linkage Agglomerative Clustering Of Amino Acid Sequences



Group Average Agglomerative Clustering Of Amino Acid Sequences





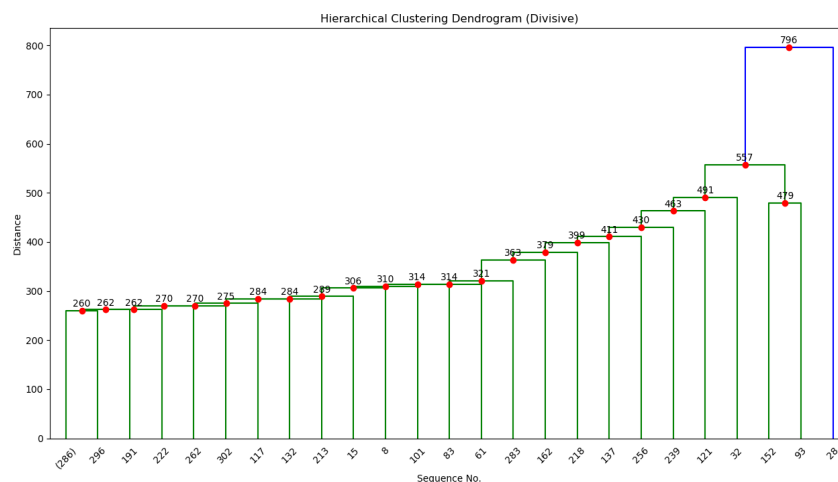
II. Divisive Clustering

We calculate the cluster with largest diameter and the select point from that cluster whose average distance from the rest of the points of that cluster, i.e. the point which is very different from most of the other points of that cluster. The point chosen is separated into a new cluster and remaining points in the cluster are rearranged. This is repeated until we have number of clusters equal to the total number of points.

Formula Used

Diameter of Cluster	$\text{Max}(d(a,b))$
---------------------	----------------------

**a and b are points in the same cluster*



III. K-Means :

We have calculated distance and formed clusters for k-value = 4. Also we have removed the outliers after calculating the distance between every pair of point.

Euclidean Distance	$J(V) = \sum_{i=1}^c \sum_{j=1}^{c_i} (\ x_i - v_j\)^2$
Centroid	$v_i = (1 / c_i) \sum_{j=1}^{c_i} x_i$

OUTPUTS:

K=4

Run #01
Initial Centroids = [267, 190, 265, 170]

Clusters Formed
Cluster 1: Centroid - 267; Cluster Size - 14
Cluster 2: Centroid - 190; Cluster Size - 22
Cluster 3: Centroid - 265; Cluster Size - 83
Cluster 4: Centroid - 9; Cluster Size - 192

Run #02
Initial Centroids = [0, 282, 147, 131]

Clusters Formed
Cluster 1: Centroid - 82; Cluster Size - 95
Cluster 2: Centroid - 282; Cluster Size - 18
Cluster 3: Centroid - 147; Cluster Size - 18
Cluster 4: Centroid - 9; Cluster Size - 180

K=6:

Run #01

Initial Centroids = [95, 4, 213, 256, 185, 279]

Clusters Formed
Cluster 1: Centroid - 235; Cluster Size - 42
Cluster 2: Centroid - 4; Cluster Size - 38
Cluster 3: Centroid - 213; Cluster Size - 22

```
Cluster 4: Centroid - 243; Cluster Size - 126
Cluster 5: Centroid - 185; Cluster Size - 49
Cluster 6: Centroid - 279; Cluster Size - 34
```

```
-----
Run #02
```

```
Initial Centroids = [248, 2, 75, 49, 122, 306]
```

```
Clusters Formed
```

```
Cluster 1: Centroid - 248; Cluster Size - 5
Cluster 2: Centroid - 9; Cluster Size - 230
Cluster 3: Centroid - 75; Cluster Size - 11
Cluster 4: Centroid - 49; Cluster Size - 45
Cluster 5: Centroid - 122; Cluster Size - 1
Cluster 6: Centroid - 306; Cluster Size - 19
-----
```

```
Run #03
```

```
Initial Centroids = [305, 244, 62, 37, 300, 42]
```

```
Clusters Formed
```

```
Cluster 1: Centroid - 9; Cluster Size - 187
Cluster 2: Centroid - 244; Cluster Size - 17
Cluster 3: Centroid - 62; Cluster Size - 9
Cluster 4: Centroid - 37; Cluster Size - 58
Cluster 5: Centroid - 300; Cluster Size - 30
Cluster 6: Centroid - 223; Cluster Size - 10
-----
```

IV. Comparison of Agglomerative and Divisive Clustering and K-Means Approach :

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 K-Means , the output is sensitive to outliers for K-6 as seen in run 4.