Hierarchical Clustering Algorithms

Dr. Sharu Theresa Jose

University of Birmingham

February 26, 2024

Learning Outcomes

- Understand the difference between hierarchical and partitional clustering algorithms.
- Apply hierarchical clustering to problems and visualise the results.
- Interpret the obtained clustering structure

Overview of Lecture

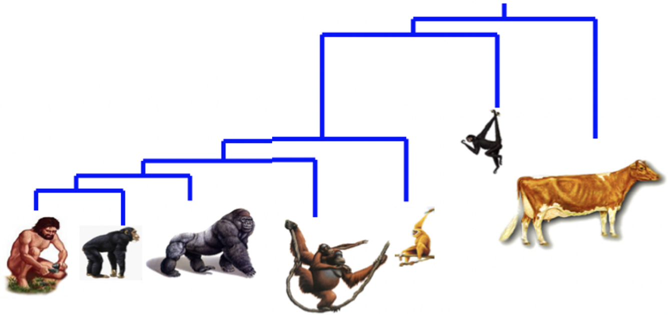
- Introduction to Hierarchical Clustering
- Agglomerative Hierarchical clustering
- Inter-Cluster Dissimilarity Metrics
- Characteristics of Hierarchical Clustering

Introduction to Hierarchical Clustering

- Recall: K-Means algorithm requires the user to supply the number K of required clusters and an initial choice of centroids.
- Hierarchical clustering requires no such specifications.
- Instead, user is only required to specify a measure of similarity (or dissimilarity) between a pair of clusters.

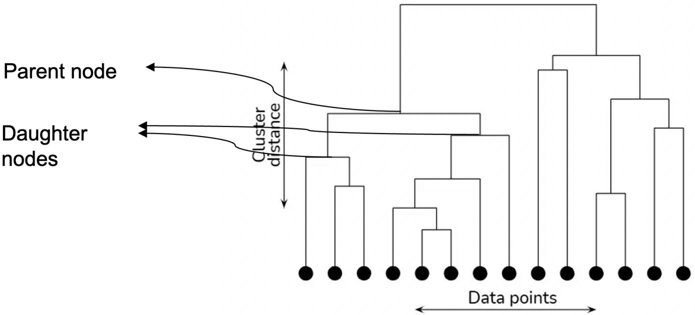
What is Hierarchical Clustering?

- Creates a hierarchical decomposition of the set of examples using a user-specified criterion
- Produces a **dendrogram**



Dendrogram

- Highly interpretable complete description of the hierarchical clustering in a graphical format
- Representation of hierarchical clustering as a rooted binary tree
- Nodes of the trees represent clusters



Pic:Prasad pai

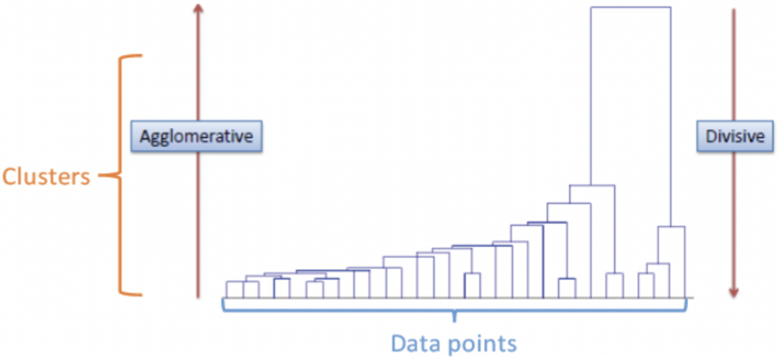
Strategies for Hierarchical Clustering

Agglomerative Clustering

- Bottom-up approach
- Starts at the bottom with each cluster containing a single observation
- At each level up, recursively merge pair of clusters with the smallest inter-cluster dissimilarity into a single cluster.
- A single cluster at the top level

Divisive Clustering

- Top-down approach
- Starts at the top with a single cluster of all observations
- At each level down, recursively split one of the existing clusters into two new clusters with the largest inter-cluster dissimilarity.
- At the bottom, each cluster contains single observation


Agglomerative
Divisive
Clusters
Data points

Agglomerative Clustering Algorithm

Algorithm

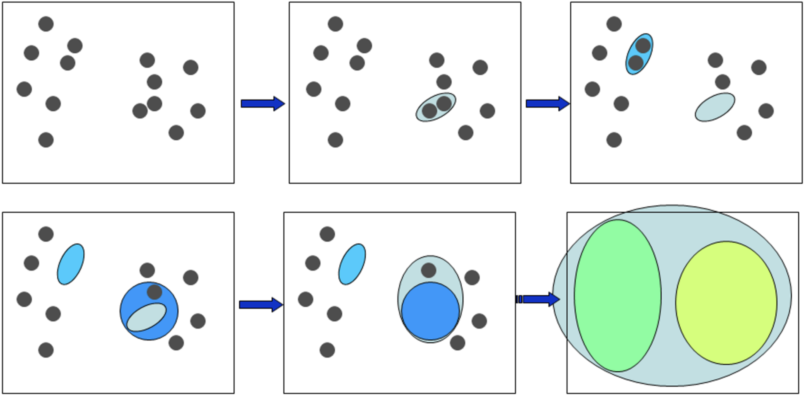
Input: For N examples, an $N \times N$ distance matrix summarizing the distance between each pair of examples.

Input: A user-specified measure of inter-cluster dissimilarity $d(C_i, C_j)$ between two clusters C_i and C_j.

1. Start with N clusters each consisting of one example.
2. Repeat until only one cluster remains:
    1. Find 2 clusters C_1 and C_2 that are most similar, or equivalently, have the smallest inter-cluster dissimilarity $d(C_1, C_2)$.
    2. Merge C_1 and C_2 into one cluster.

Output: A Dendrogram

An Illustration



Input 1: Distance Matrix

What is distance matrix?

- Given N observations $(\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(N)})$ of examples/feature vectors, **distance matrix** summarizes the similarity relationship among the N observations.
- Distance matrix D is an $N \times N$ matrix (matrix with N rows and N columns) whose entry in ith row and jth column is given by

$$D_{i,j} = d(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}),$$

  where $d(\mathbf{x}^{(i)}, \mathbf{x}^{(j)})$ is a given distance measure (e.g., Euclidean, Manhattan, Chebychev etc.).
- Properties of distance matrix:
  - Symmetric: $D_{i,j} = D_{j,i}$
  - Zero-diagonal entries: $D_{i,i} = 0$.

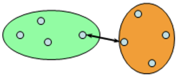
Input 2: Inter-Cluster Dissimilarity Metrics

Single Linkage (SL)

SL distance is the shortest distance from any member of the cluster to any member of the other cluster. For two clusters C_1 and C_2,

$$d_{\mathrm{SL}}(C_1, C_2) = \min_{i \in C_1, j \in C_2} d(i, j),$$

where $d(i, j)$ denotes a distance measure (eg., Euclidean, Manhattan etc.) between example i in cluster C_1 and example j in cluster C_2.



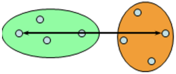
- Agglomerative clustering with single-linkage merges two clusters with the smallest single-linkage distance between them in each step.

Input 2: Inter-Cluster Dissimilarity Metrics

Complete Linkage (CL)

CL distance is the largest distance from any member of the cluster to any member of the other cluster:

$$d_{\mathrm{CL}}(C_1, C_2) = \max_{i \in C_1, j \in C_2} d(i, j).$$



- Agglomerative clustering with complete-linkage merges two clusters with the smallest complete-linkage distance between them in each step.

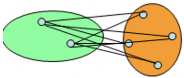
Input 2: Inter-Cluster Dissimilarity Metrics

Group Average (GA)

GA distance is the average distance between members of the two clusters:

$$d_{\mathrm{GA}}(C_1, C_2) = \frac{1}{N_{C_1} N_{C_2}} \sum_{i \in C_1, j \in C_2} d(i, j),$$

where N_{C_1} and N_{C_2} denote the number of examples in cluster C_1 and C_2 respectively.



- Agglomerative clustering with group average merges two clusters with the smallest group average distance between them in each step.

Comparison between the linkages

Single Linkage

- SL is determined by the pair of examples in the two clusters that are the closest; dissimilarities between other pairs of examples in the clusters do not matter.
- Consequently, SL induces **chaining effect**: tendency to combine clusters linked by a series of close intermediate examples.
- Results in clusters that are not compact

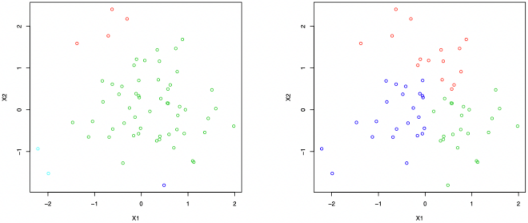
Complete Linkage

- Requires all examples in the two clusters to be relatively similar
- Produces compact clusters with small diameters
- However, an example in a CL-linkage based merged cluster can be closer to examples in other clusters than to examples in its own cluster. This induces **crowding** of clusters with clusters not far enough apart.

Group Average

- Attempts to produce relatively compact clusters that are relatively far apart
- Results of group average clustering can change with a monotone increasing transformation of the distance measures (that is, if we changed the distance, but maintained the ranking of the distances, the cluster solution could change).

Illustration of Chaining and Crowding



Single linkage   Complete linkage

Example 1: Clustering of European Cities Based on Air Distance

Given the distance matrix as below, obtain a **single-linkage dendrogram**.

	Lond	Paris	Berlin	Prague	Zurich	Milan
Lond	0	393	932	1027	776	958
Paris		0	878	883	489	641
Berlin			0	279	650	795
Prague				0	528	401
Zurich					0	204
Milan						0

Solution

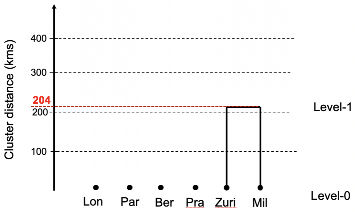
Level 0:

Clusters: (Lond), (Paris), (Berlin), (Prague), (Zurich), (Milan)

Level 1:

	Lond	Paris	Berlin	Prague	Zurich	Milan
Lond	0	393	932	1027	776	958
Paris		0	878	883	489	641
Berlin			0	279	650	795
Prague				0	528	401
Zurich					0	204
Milan						0

- Clusters (Milan) and (Zurich) have the smallest SL distance of 204. Merge them.
- New Clusters: (Lond), (Paris), (Berlin), (Prague), (Zurich, Milan)



In the dendrogram, height at which two clusters merge corresponds to their inter-cluster dissimilarity distance.

Level 2:

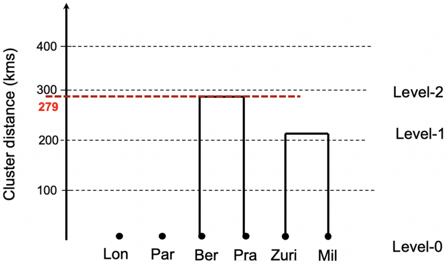
	Lond	Paris	Berlin	Prague	(Zurich, Milan)
Lond	0	393	932	1027	?
Paris		0	878	883	?
Berlin			0	279	?
Prague				0	?
(Zurich, Milan)					0

Compute the following SL distances:

$$d_{\mathrm{SL}}(Lond, (Zurich, Milan)) = \min\{d(Lond, Zurich), d(Lond, Milan)\} = \min\{776, 958\} = 776$$
$$d_{\mathrm{SL}}(Paris, (Zurich, Milan)) = \min\{d(Paris, Zurich), d(Paris, Milan)\} = \min\{489, 641\} = 489$$
$$d_{\mathrm{SL}}(Berlin, (Zurich, Milan)) = \min\{d(Berlin, Zurich), d(Berlin, Milan)\} = \min\{650, 795\} = 650$$
$$d_{\mathrm{SL}}(Prague, (Zurich, Milan)) = \min\{d(Prague, Zurich), d(Prague, Milan)\} = \min\{528, 401\} = 401$$

	Lond	Paris	Berlin	Prague	(Zurich, Milan)
Lond	0	393	932	1027	776
Paris		0	878	883	489
Berlin			0	279	650
Prague				0	401
(Zurich, Milan)					0

- Clusters (Berlin) and (Prague) have the smallest SL linkage distance (=279). Merge them.
- New clusters: (Lond), (Paris), (Berlin, Prague), (Zurich, Milan)



Level 3:

	Lond	Paris	(Berlin, Prague)	(Zurich, Milan)
Lond	0	393	?	776
Paris		0	?	489
(Berlin, Prague)			0	?
(Zurich, Milan)				0

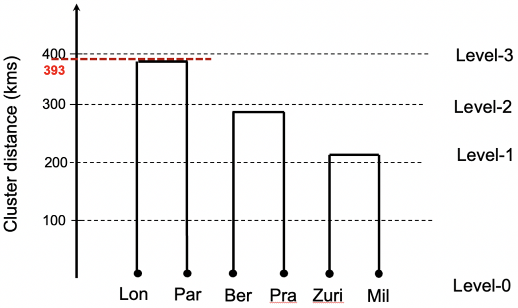
The SL distances can be computed as:

$$d_{\mathrm{SL}}(Lond, (Berlin, Prague)) = \min\{932, 1027\} = 932$$
$$d_{\mathrm{SL}}(Paris, (Berlin, Prague)) = \min\{878, 883\} = 878$$
$$d_{\mathrm{SL}}((Zurich, Milan), (Berlin, Prague)) = \min\{650, 528, 795, 401\} = 401.$$

This results in:

	Lond	Paris	(Berlin, Prague)	(Zurich, Milan)
Lond	0	393	932	776
Paris		0	878	489
(Berlin, Prague)			0	401
(Zurich, Milan)				0

- Clusters (Lond) and (Paris) have the smallest SL linkage. Merge them.
- New clusters: (Lond,Paris), (Berlin, Prague), (Zurich, Milan)

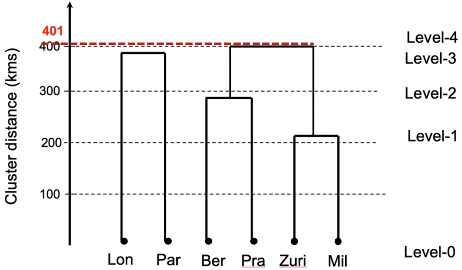


Level 4:

For the new clusters, again compute the SL distances. This results in the following matrix.

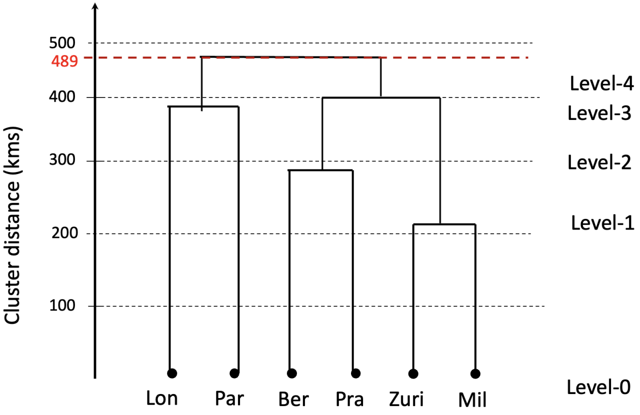
	(Lond, Paris)	(Berlin, Prague)	(Zurich, Milan)
(Lond,Paris)	0	878	489
(Berlin, Prague)		0	401
(Zurich, Milan)			0

- Clusters (Berlin, Prague) and (Zurich, Milan) have the smallest SL distance (=401). Merge them.
- New clusters: (Lond, Paris), (Berlin, Prague, Zurich, Milan)



Level 5:

- Finally, the two clusters (Lond, Paris) and (Berlin, Prague, Zurich, Milan) merge.
- The height at which they merge is given by the $d_{\mathrm{SL}}((Lond, Paris), (Berlin, Prague, Zurich, Milan)) = 489$.



Homework Question

For the same example, try implementing complete linkage and group average agglomerative clustering algorithms. Compare the obtained dendrograms.

Reading a Dendrogram

- Height at which two clusters merge corresponds to their inter-cluster dissimilarity distance
- Possesses a monotonicity property, i.e., inter-cluster dissimilarity between merged clusters is monotonically increasing with the level of the merger.
- Horizontally cutting dendrogram at a particular height partitions observations into disjoint clusters.

Space and Time Complexity

- Storage Complexity: $O(N^2)$
  - Storing the distance matrix requires storage of $N^2/2$ entries.
- Time Complexity is $O(N^3)$ in many cases.
  - There are N iterations, and in each iteration the N^2-size distance matrix needs to be updated and searched.
  - Complexity can be reduced to $O(N^2 \log(N))$ time for some approaches.
- Space and time complexity severely limits the size of datasets that can be processed.

Characteristics of Hierarchical Clustering

- Lack of a global objective function
  - Need not solve hard combinatorial optimization problem as in K-means
  - No issues with local minima or choosing initial points
- Deterministic algorithm
- Merging decisions are final. But may impose a hierarchical structure on an otherwise un-hierarchical data.