DM583

Data Mining

Hierarchical Clustering

Why Hierarchical Clustering ?

- Oftentimes, we don't want to look at a single partition of the dataset with *k* clusters, but rather explore a whole spectrum of partitions at different levels of granularity, corresponding to different numbers of clusters
- In other words, we may want a **clustering hierarchy**, rather than a single partition
- The reasons are manyfold:
  - By building a hierarchy one doesn't need to specify the number of clusters *k* in advance
  - Rather, a hierarchical structure may help determine the best *k* (if any) a posteriori
  - The hierarchical structure may also reveal that the data is naturally organised hierarchically
    - sub-clusters inside clusters
  - The hierarchy provides a powerful **visualisation** tool of high-dimensional data
    - which may reveal not only clusters and sub-clusters, but also potential **outliers**

Clustering Hierarchies

- **Hierarchy** (of data partitions):
  - **Sequence of nested hard partitions**
    - A partition $\mathbf{P}_1$ is *nested* within $\mathbf{P}_2$ if each component (cluster) of $\mathbf{P}_1$ is a subset of a component of $\mathbf{P}_2$
- **Example:**

$\mathbf{P}_1 = \{ (\mathbf{x}_1), (\mathbf{x}_3, \mathbf{x}_4, \mathbf{x}_6), (\mathbf{x}_2, \mathbf{x}_5) \}$

$\mathbf{P}_2 = \{ (\mathbf{x}_1, \mathbf{x}_3, \mathbf{x}_4, \mathbf{x}_6), (\mathbf{x}_2, \mathbf{x}_5) \}$

- **Counter-Example:**

$\mathbf{P}_3 = \{ (\mathbf{x}_1, \mathbf{x}_3, \mathbf{x}_4, \mathbf{x}_6), (\mathbf{x}_2, \mathbf{x}_5) \}$

$\mathbf{P}_4 = \{ (\mathbf{x}_1, \mathbf{x}_2), (\mathbf{x}_3, \mathbf{x}_4, \mathbf{x}_6), (\mathbf{x}_5) \}$

Clustering Hierarchies

- A *complete hierarchy*:
  - Starts or ends with a completely *disjoint partition/clustering*
    - ***Disjoint clustering***: contains only **atomic clusters** (***singletons***)
    - Example: $\mathbf{P} = \{ (\mathbf{x}_1), (\mathbf{x}_2), (\mathbf{x}_3), (\mathbf{x}_4), (\mathbf{x}_5), (\mathbf{x}_6) \}$
      - It is also called "trivial clustering solution"
  - Starts or ends with a single (not partitioned) component
    - Dataset itself as a single "cluster" with all data objects
    - Example: $\mathbf{P} = \{ (\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \mathbf{x}_4, \mathbf{x}_5, \mathbf{x}_6) \}$
  - Generally, there are ***N*** – 2 intermediate partitions

- **Hierarchies** are commonly used to organise information:
  - Categories and subcategories
  - Example: Phylogenetic Trees in Biology

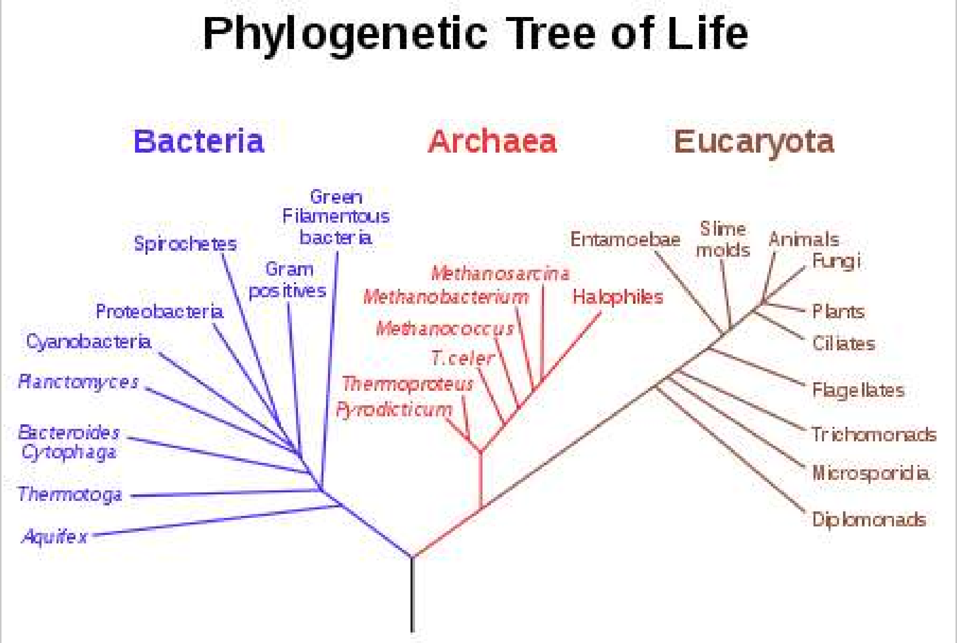
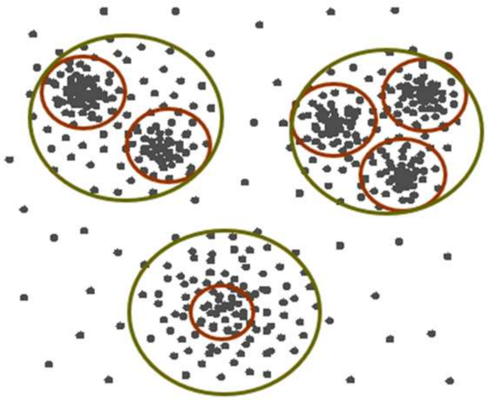
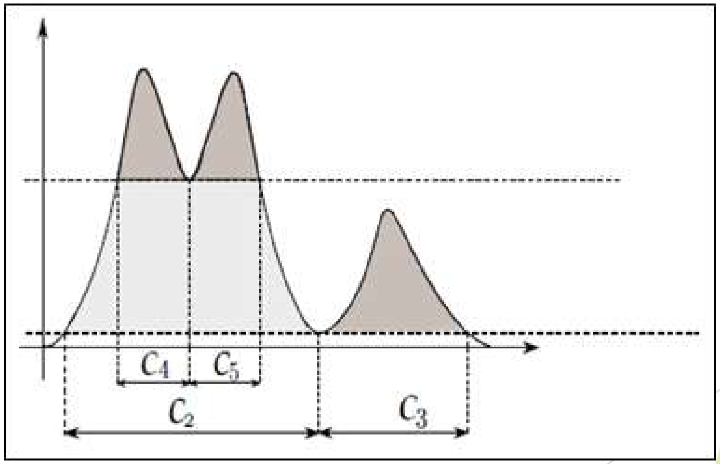


Fig. 1: A speculatively rooted tree for rRNA genes

http://en.wikipedia.org/wiki/Phylogenetic_tree

- The relationship between natural clusters and subclusters in many datasets is intrinsically **hierarchical**:
    - Examples:





Hierarchical Clustering as a Relational Approach

Hierarchical Clustering can operate with (dis)similarities only: they are (or can be turned into) **relational algorithms**

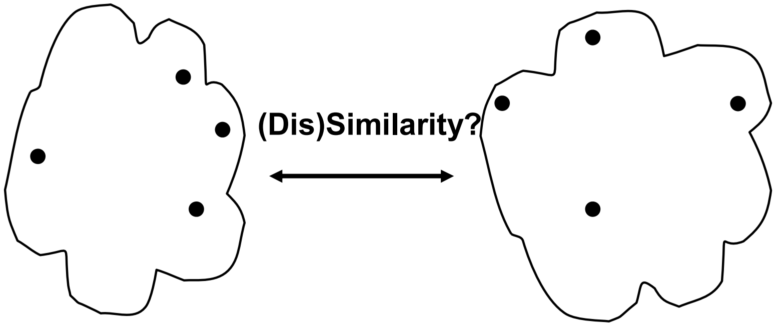
0	8	8	7	7
	0	2	4	4
		0	3	3
			0	1
				0

Classic Agglomerative Hierarchical Clustering Algorithm (AHC)

1. Start with each observation being a cluster on its own (i.e., a **singleton**), and compute all pairwise distances between observations (if not given as input)
2. Find the **closest pair** of current clusters and merge them into a single cluster
3. Compute the **distance** between this newly born cluster and the other clusters
4. Repeat Steps 2 and 3 until a single cluster remains

There is a number of algorithms that follow exactly the same steps above. The difference between them is how the **distances between clusters** are computed

How to Define Inter-Cluster (Dis)Similarity



	p1	p2	p3	p4	p5	. . .
p1						
p2						
p3						
p4						
p5						

.

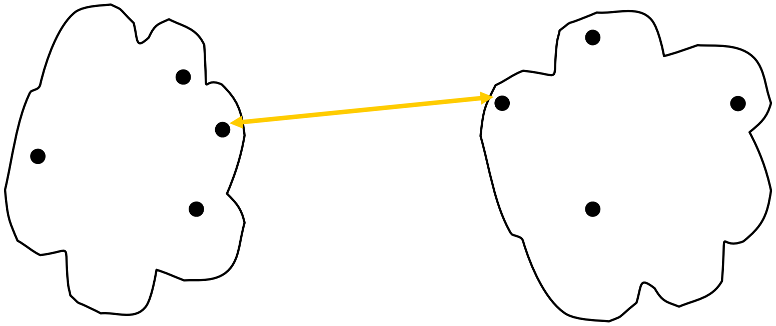
.

.

Proximity Matrix

- MIN
- MAX
- Average
- Distance Between Centroids
- Other methods
  - Ward's
  - …

How to Define Inter-Cluster (Dis)Similarity

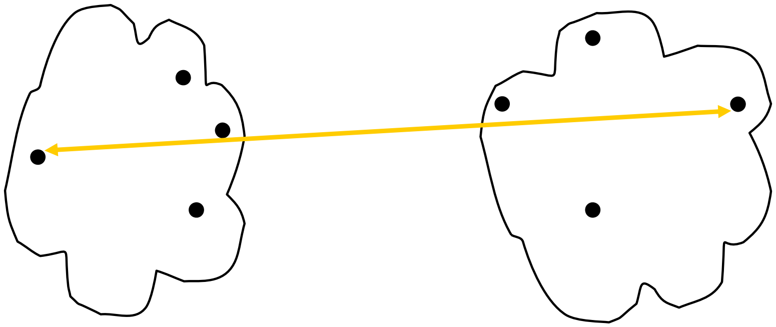


	p1	p2	p3	p4	p5	. . .
p1						
p2						
p3						
p4						
p5						
.						
.						
.						

Proximity Matrix

- MIN
- MAX
- Average
- Distance Between Centroids
- Other methods
  - Ward's
  - …

How to Define Inter-Cluster (Dis)Similarity

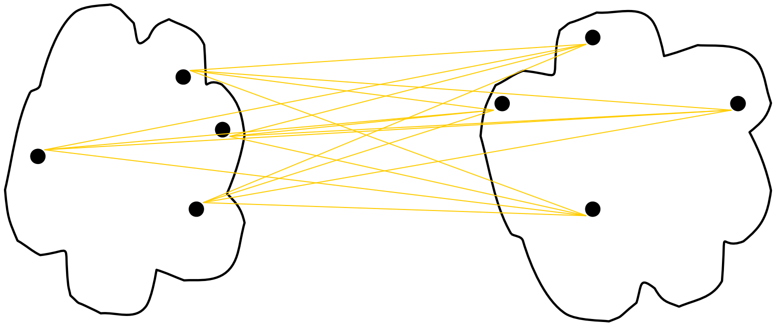


	p1	p2	p3	p4	p5	. . .
p1						
p2						
p3						
p4						
p5						
.						
.						
.						

Proximity Matrix

- MIN
- MAX
- Average
- Distance Between Centroids
- Other methods
  - Ward's
  - …

How to Define Inter-Cluster (Dis)Similarity

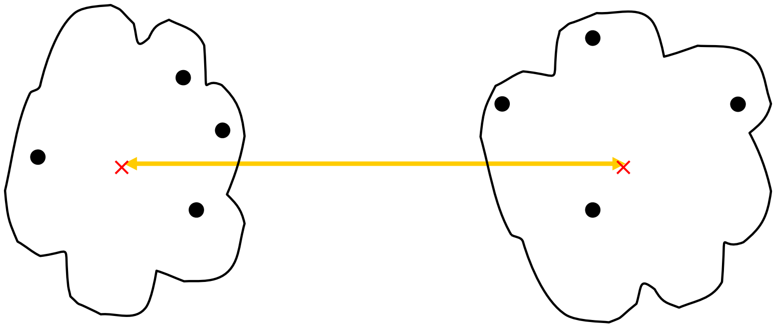


	p1	p2	p3	p4	p5	. . .
p1						
p2						
p3						
p4						
p5						
.						
.						
.						

Proximity Matrix

- MIN
- MAX
- Average
- Distance Between Centroids
- Other methods
  - Ward's
  - …

How to Define Inter-Cluster (Dis)Similarity



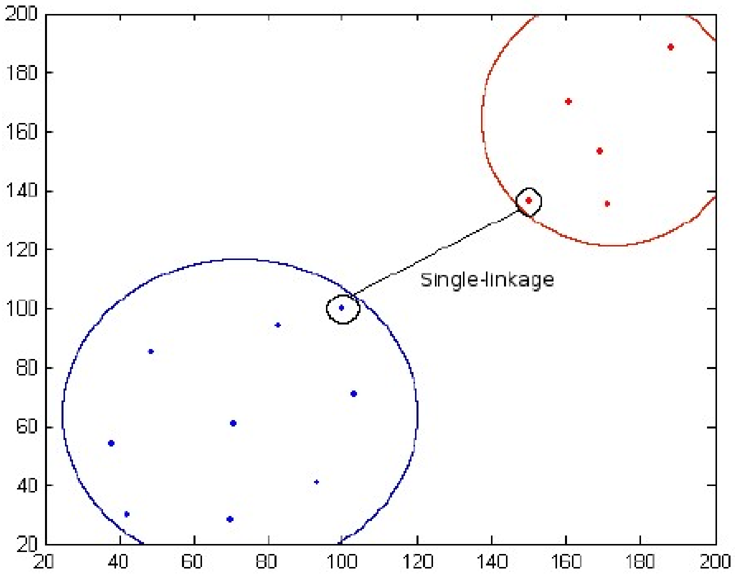
	p1	p2	p3	p4	p5	. . .
p1						
p2						
p3						
p4						
p5						
.						
.						
.						

Proximity Matrix

- MIN
- MAX
- Average
- Distance Between Centroids
- Other methods
  - Ward's
  - …

Agglomerative Hierarchical Clustering Algorithm

- **Single-Linkage**:
  - Distance between two clusters is measured as the smallest distance between any two observations



Computation of Single-Linkage

- Useful property of min:
  - min{**D**} = min{ min{$\mathbf{D}_1$} , min{$\mathbf{D}_2$} }
    - **D**, $\mathbf{D}_1$ and $\mathbf{D}_2$ are real-valued sets such that $\mathbf{D}_1 \cup \mathbf{D}_2 = \mathbf{D}$
  - Example:
    - min{10, -3, 0, 100} = min { min{10, -3}, min{0, 100} } = -3
  - Property holds recursively (for min{$\mathbf{D}_1$} and min{$\mathbf{D}_2$})
- Why can this property be useful for Single-Linkage ?
  - Given the distances between a cluster **A** and two clusters **B** and **C** that have been merged
    - It is trivial to compute the distance between **A** and ($\mathbf{B} \cup \mathbf{C}$) from the previous distances

Single-Linkage (Example):

- Initial distance matrix ($\mathbf{D}_1$) for 5 observations {1, 2, 3, 4, 5}: first merge takes place between singletons 1 and 2 (closest clusters)

$$\mathbf{D}_1 = \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} \begin{bmatrix} 0 & & & & \\ \boxed{2} & 0 & & & \\ 6 & 5 & 0 & & \\ 10 & 9 & 4 & 0 & \\ 9 & 8 & 5 & 3 & 0 \end{bmatrix}$$

- Updating the distances (between the new cluster and the others):

$d_{(12)3}=min\{d_{13},d_{23}\}=d_{23}=5;$

$d_{(12)4}=min\{d_{14},d_{24}\}=d_{24}=9;$

$d_{(12)5}=min\{d_{15},d_{25}\}=d_{25}=8;$

- The result is a new distance matrix ($\mathbf{D}_2$)

$$\mathbf{D}_2 = \begin{array}{r} 12 \\ 3 \\ 4 \\ 5 \end{array}\begin{bmatrix} 0 & & & \\ 5 & 0 & & \\ 9 & 4 & 0 & \\ 8 & 5 & \boxed{3} & 0 \end{bmatrix}$$

- The closest clusters now are (singletons) 4 and 5
- Merging **4** and **5** yields three clusters: {1,2}, {4,5}, {3}
- Since $d_{(12)3}$ is already available, we can update the distances as:

  $d_{(45)(12)} = min\{d_{4(12)}, d_{5(12)}\} = d_{5(12)} = 8$

  $d_{(45)3} = min\{d_{43}, d_{53}\} = d_{43} = 4$

  which yields the following new matrix:

$$\mathbf{D}_3 = \begin{array}{r} 12 \\ 3 \\ 45 \end{array}\begin{bmatrix} 0 & & \\ 5 & 0 & \\ 8 & \boxed{4} & 0 \end{bmatrix}$$

Merge *clusters* {3} and {4,5};

Finally, merge {3,4,5} and {1,2} into a single cluster

- **Note on Computational Speed-Up of Single Linkage**:

  The (dis)similarity between 2 clusters follows immediately from the (dis)similarity matrix updated in the previous iteration, such that *there is no need to resort to the original matrix*

  - For instance, in the previous example we simplified the computation of $d_{(12)(45)}$ as min{$d_{(12)(4)}$, $d_{(12)(5)}$} by making use of the min property:
    - min{$d_{(12)(4)}$, $d_{(12)(5)}$} = min {9, 8} = min{d_{14}, d_{24}, d_{15}, d_{25}}

- The sequence of nested partitions obtained in the previous example was:

  { (1), (2), (3), (4), (5) } → { (1, 2), (3), (4), (5) } →

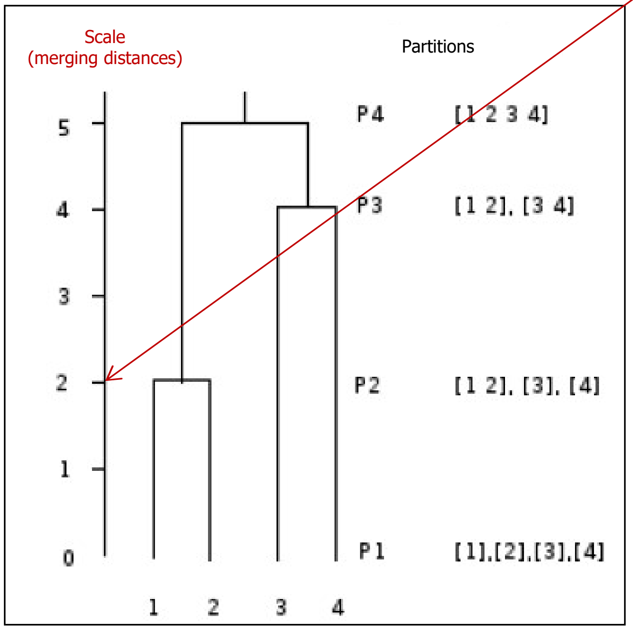
  { (1, 2), (3), (4, 5) } → { (1, 2), (3, 4, 5) } → { (1, 2, 3, 4, 5) }

- With this collection of nested partitions, as well as the distances corresponding to the pairs of merged clusters, we can build a **dendrogram**
  - Dendrogram = visual representation of the **clustering hierarchy** enhanced with a **scale** of critical distances between clusters
  - A powerful tool for data *visualisation* and *exploratory data analysis*

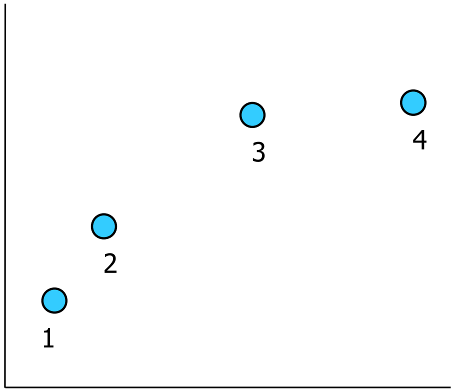
Dendrogram (Example):

$$\mathbf{D} = \begin{array}{c|cccc} 1 & 0 & 2 & 7 & 13 \\ 2 & 2 & 0 & 5 & 10 \\ 3 & 7 & 5 & 0 & 4 \\ 4 & 13 & 10 & 4 & 0 \end{array}$$

merging distances
(between **clusters**)



Dendrogram



dataset

Exercise

- Draw the complete **dendrogram** (hierarchy + vertical axis/scale) for one of our previous examples of Single-Linkage:

$$\mathbf{D}_1 = \begin{array}{c@{}c} \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} & \begin{bmatrix} 0 & & & & \\ 2 & 0 & & & \\ 6 & 5 & 0 & & \\ 10 & 9 & 4 & 0 & \\ 9 & 8 & 5 & 3 & 0 \end{bmatrix} \end{array}$$

Dendrograms and Partitions

Partitions can be obtained by "cutting through" the dendrogram

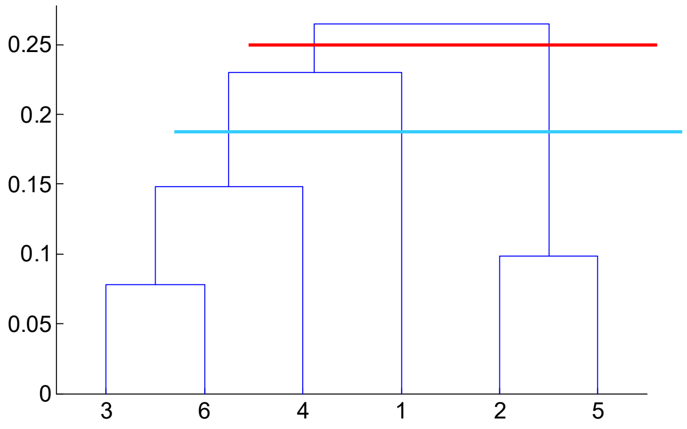
Horizontal cuts

no. of clusters = no. of line intersections

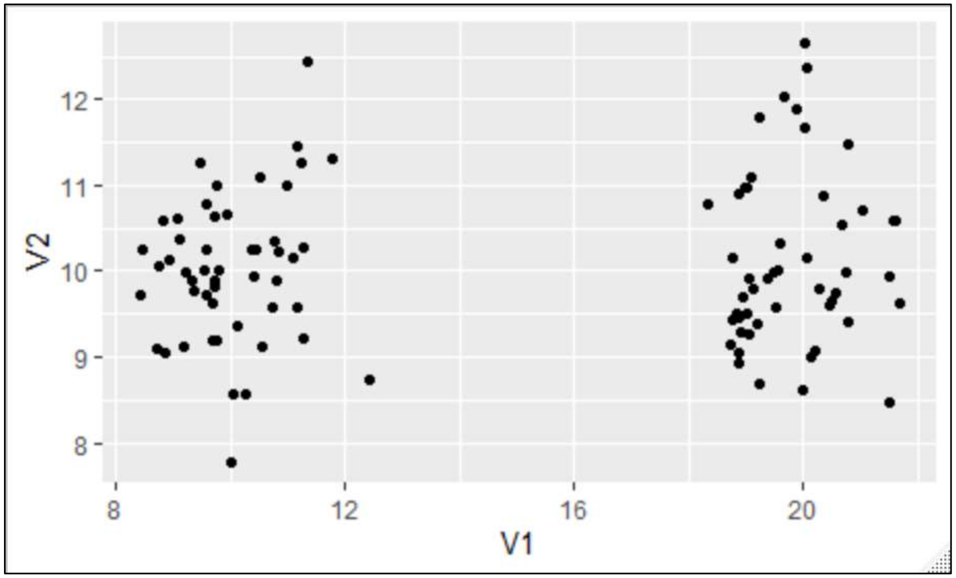
Examples:

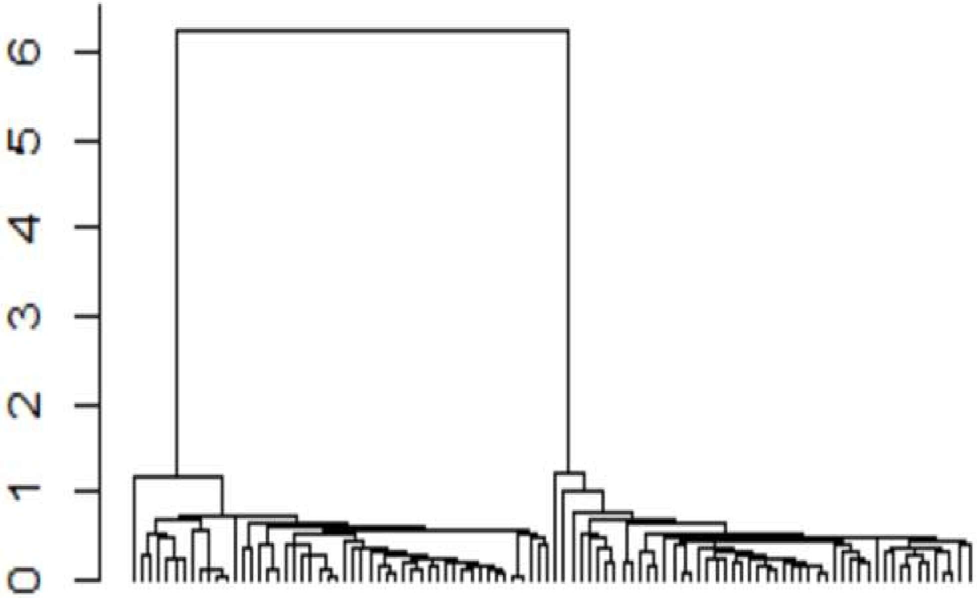
$P_2 = \{ (\mathbf{x}_1,\mathbf{x}_3,\mathbf{x}_4,\mathbf{x}_6), (\mathbf{x}_2,\mathbf{x}_5) \}$

$P_1 = \{ (\mathbf{x}_1), (\mathbf{x}_3,\mathbf{x}_4,\mathbf{x}_6), (\mathbf{x}_2,\mathbf{x}_5) \}$

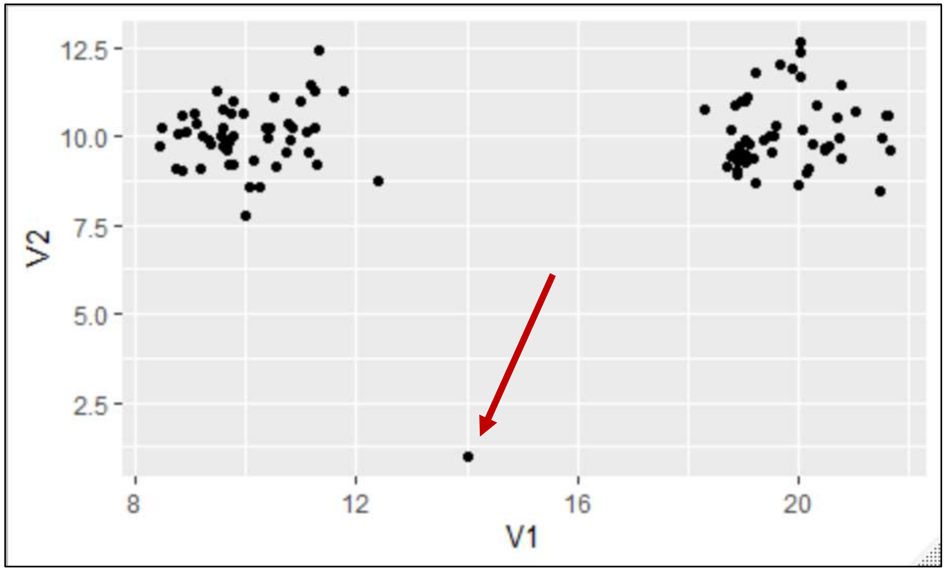


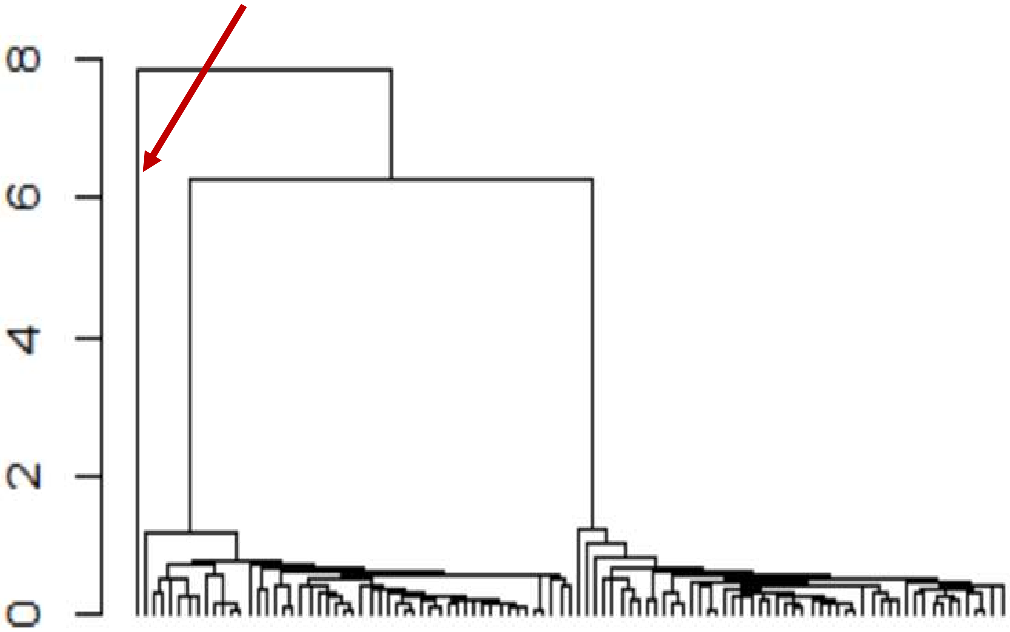
A dendrogram may visually indicate the number of natural clusters:





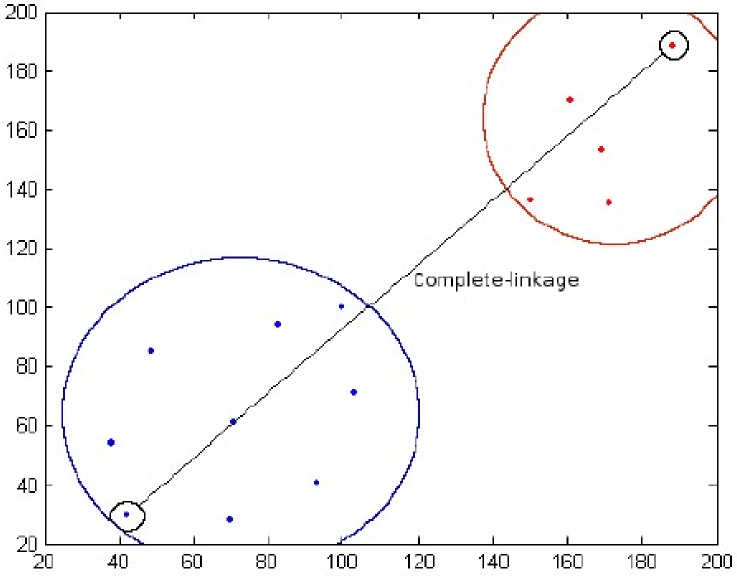
A dendrogram may also indicate the presence of outliers:





Agglomerative Hierarchical Clustering Algorithm

- **Complete-Linkage (Max)**:
  - Distance between two clusters is measured as the largest distance between any two observations



Computation of Complete-Linkage

- Useful property of max:
  - max{$\mathbf{D}$} = max{ max{$\mathbf{D}_1$} , max{$\mathbf{D}_2$} }
    - $\mathbf{D}$, $\mathbf{D}_1$ and $\mathbf{D}_2$ are real-valued sets such that $\mathbf{D}_1 \cup \mathbf{D}_2 = \mathbf{D}$
  - Example:
    - max{10, -3, 0, 100} = max { max{10, -3}, max{0, 100} } = 100
  - Property holds recursively (for max{$\mathbf{D}_1$} and max{$\mathbf{D}_2$})
- Why can this property be useful for Complete-Linkage ?
  - Given the distances between a cluster **A** and two clusters **B** and **C** that have been merged
    - It is trivial to compute the distance between **A** and (**B** $\cup$ **C**) from the previous distances

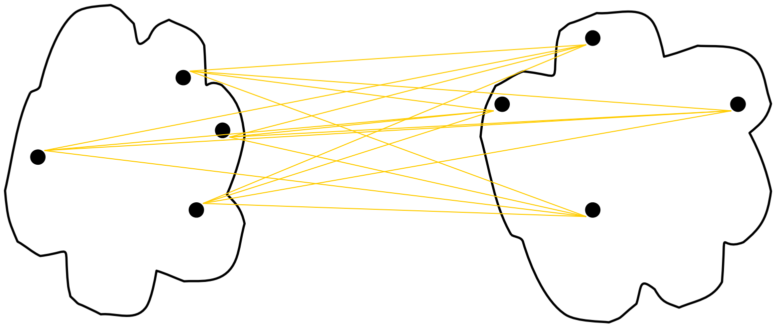
Complete-Linkage (Exercise):

- Initial distance matrix ($\mathbf{D}_1$) for 5 observations {1, 2, 3, 4, 5}

$$\mathbf{D}_1 = \begin{array}{cc} \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} & \begin{bmatrix} 0 & & & & \\ \boxed{2} & 0 & & & \\ 6 & 5 & 0 & & \\ 10 & 9 & 4 & 0 & \\ 9 & 8 & 5 & 3 & 0 \end{bmatrix} \end{array}$$

- First merge takes place between singletons 1 and 2 (closest clusters)
  - Rest is the same as for Single-Linkage, but now the distances are updated in a different way
- **Exercise:** compute the ***complete-linkage clustering*** for these data and draw the resulting ***dendrogram*** (make sure you include the height at which the cluster mergers occur, which now correspond to the complete-linkage cluster distances)

How to Define Inter-Cluster (Dis)Similarity



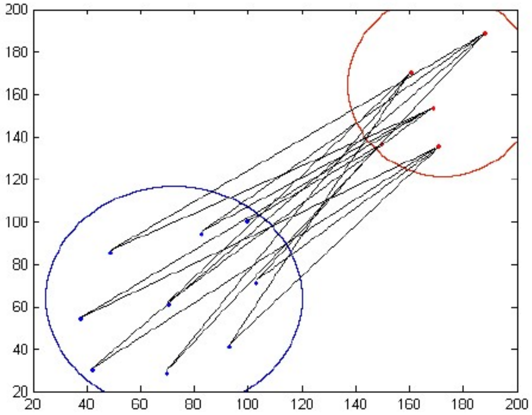
	p1	p2	p3	p4	p5	. . .
p1						
p2						
p3						
p4						
p5						
.						
.						
.						

Proximity Matrix

- MIN
- MAX
- **Average**
- Distance Between Centroids
- Other methods
  - Ward's
  - …

Average Linkage (UPGMA)

- Distance between *clusters* is given by the average distance between pairs of objects across the clusters in question
- It is also known as Group Average or **UPGMA**:
  - *Unweighted Pair Group Method using Arithmetic averages*
    - "unweighted" → every pair of objects has the same importance



Ricardo Campello

Efficient Proximity Matrix Updates

- The (dis)similarity between a newly born cluster (formed by merging two existing clusters) and other clusters can be updated from the previously computed distances
  - rather than computed from scratch
- Specifically, let $|\mathbf{C}_i|$ be the number of objects in a cluster $\mathbf{C}_i$ and $d(\mathbf{C}_i,\mathbf{C}_j)$ be the (dis)similarity between two clusters $\mathbf{C}_i$ and $\mathbf{C}_j$. One can show that:

$$\boxed{d(\mathbf{C}_i, \mathbf{C}_j \cup \mathbf{C}_k) = \frac{|\mathbf{C}_j|}{|\mathbf{C}_j| + |\mathbf{C}_k|} d(\mathbf{C}_i, \mathbf{C}_j) + \frac{|\mathbf{C}_k|}{|\mathbf{C}_j| + |\mathbf{C}_k|} d(\mathbf{C}_i, \mathbf{C}_k)}$$

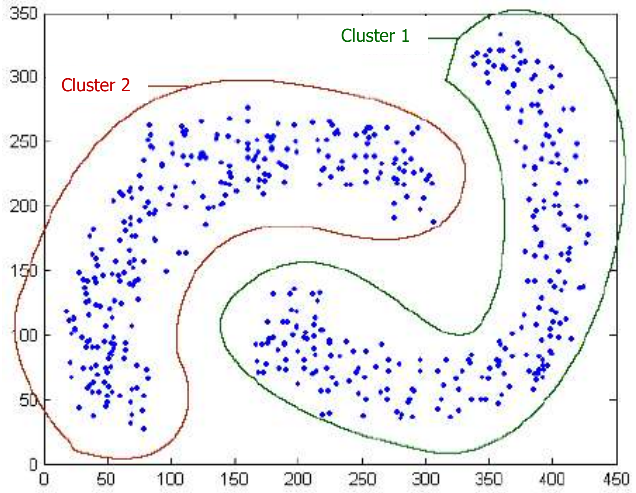
Exercise:

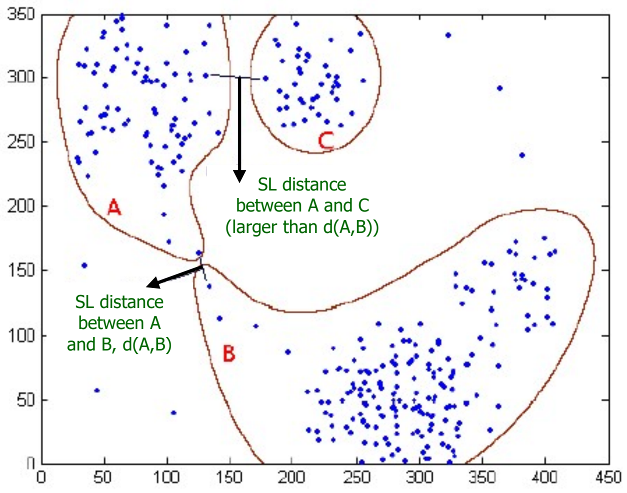
- Produce the complete dendrogram resulting from the application of the average linkage (**UPGMA**) method to the distance matrix below
    - Show, step-by-step, the updated distance matrix (using the formula in the previous slide)

$$\mathbf{D}_1 = \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} \begin{bmatrix} 0 & & & & \\ 2 & 0 & & & \\ 6 & 5 & 0 & & \\ 10 & 9 & 4 & 0 & \\ 9 & 8 & 5 & 3 & 0 \end{bmatrix}$$

Summary of Single-, Complete-, and Average-Linkage

- Single-Linkage:
  - It is capable of detecting clusters with arbitrary shapes (left figure)
  - However, it is not robust to noisy data (figure on the right)





Summary of Single-, Complete-, and Average-Linkage

- Complete Linkage:
  - Less sensitive to noisy data, but
    - It can be sensitive to outliers
    - It is more prone to split large clusters (even if they are "legitimate" clusters)
    - It cannot detect stretched clusters with arbitrary shapes, close to each other
- Average Linkage:
  - It tends to be less sensitive to noisy data and outliers than the previous methods
  - But like complete-linkage, it tends to favour "globular-shaped" clusters
  - It is usually a good compromise in practice
    - Alongside with more sophisticated methods, such as Ward's and density-based algorithms (e.g. HDBSCAN*)

Ward's Method (1963)

- This method is based on the successive minimisation of the Sum of Squared Errors – SSE (cluster within variances) at each new hierarchical level (partition) as built bottom-up:

$$J = \sum_{i=1}^{k} \sum_{\mathbf{x}_j \in \mathbf{C}_i} d\left(\mathbf{x}_j, \overline{\mathbf{x}}_i\right)^2$$

where d = Euclidian distance and $\overline{\mathbf{x}}_i$ is the i-th cluster centre:

$$\overline{\mathbf{x}}_i = \frac{1}{\left|\mathbf{C}_i\right|} \sum_{\mathbf{x}_i \in \mathbf{C}_i} \mathbf{x}_i$$

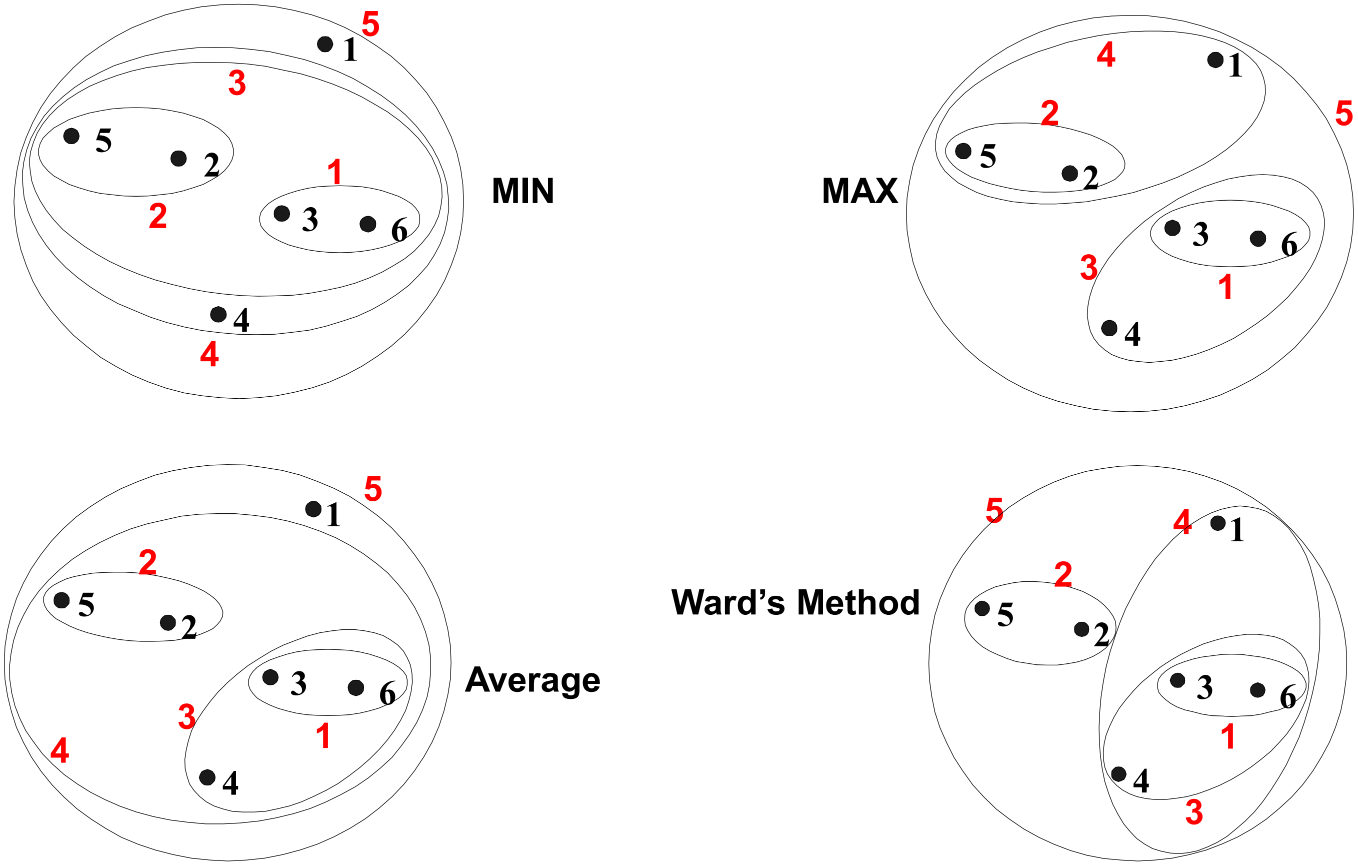
Ward’s Method

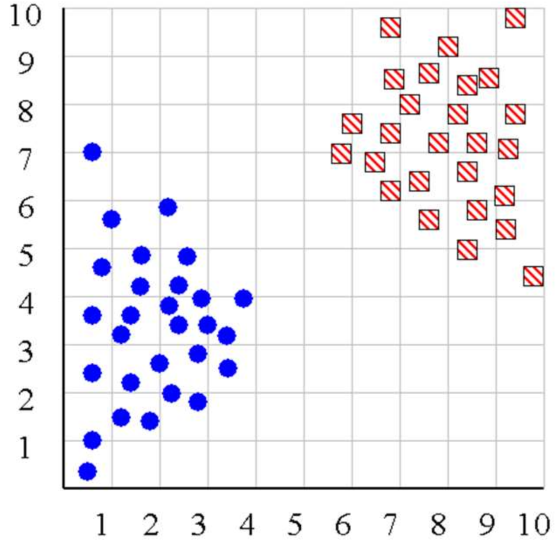
- “Dissimilarity” between pairs of clusters **C**$_i$ and **C**$_j$:
  - is defined as the variation that would be observed in the SSE (J criterion) of the current partition should these clusters be merged to form the next partition
  - thus, merging the two most similar clusters is equivalent to minimising the increase in within-cluster variances at each new hierarchical level

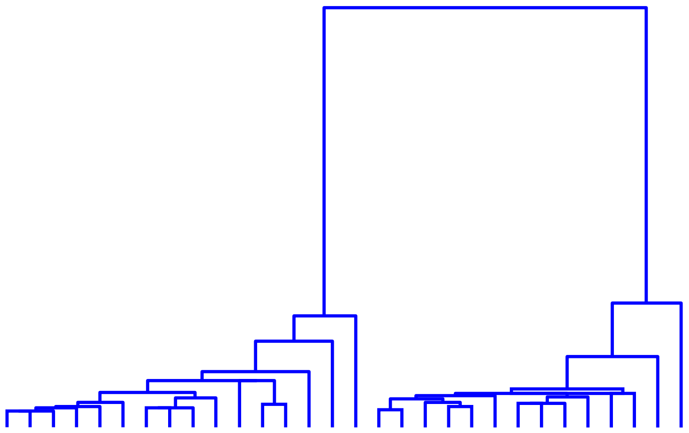
Ward’s Method

- **Cons**:
  - Like Average Linkage, it tends to produce globular clusters
  - Update formula is only interpretable for real-valued vector data and Euclidean distance
- **Pros:**
  - Similar to Average Linkage regarding robustness to noise/outliers
  - “Hierarchical Counterpart” of k-means (same objective function)
    - It can be used to initialise k-means
  - **Jain & Dubes (1988):** “*Several of the comparative studies discussed in Section 3.5.2 conclude that Ward’s method, also called the minimum variance method, outperforms other hierarchical clustering methods*”

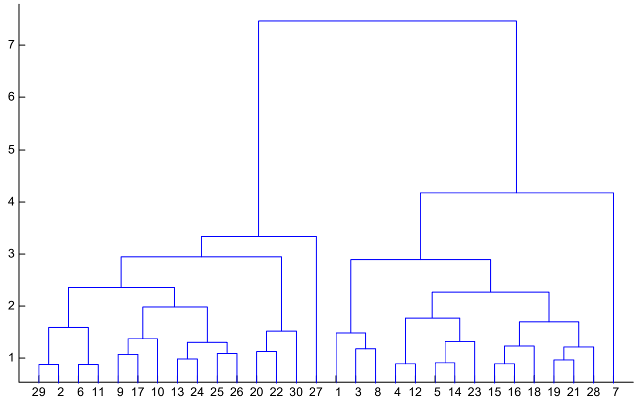
Hierarchical Clustering: Comparison



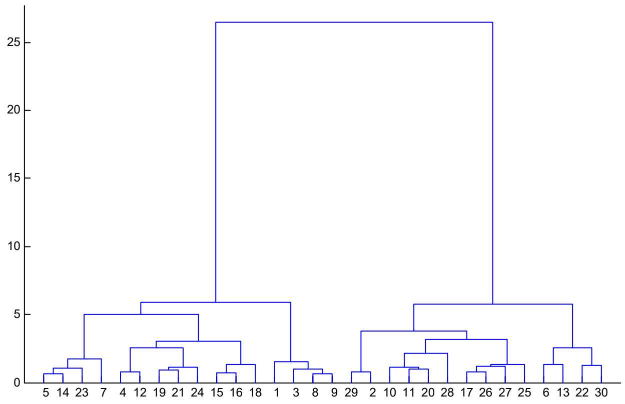




Single Linkage



Average Linkage



Ward's

Keogh, E. A Gentle Introduction to Machine Learning and Data Mining for the Database Community, SBBD 2003, Brazil.

Hierarchical Clustering: Typical Time and Space Requirements

- $O(N^2)$ **memory** since it uses the proximity matrix
  - N is the number of points
- $O(N^3)$ **runtime** in many cases
  - There are N steps and, at each step, the proximity matrix must be updated and searched
- Complexity can be reduced for some approaches, e.g., Single-Linkage and Complete-Linkage

Summary of Hierarchical Methods

- **No. of Clusters**: we don't need to specify the number of clusters *in advance*, but we may need to do it *a posteriori* ...

- **Greedy Procedure**: it is not possible to fix a wrong merger in previous steps – optimal solution (e.g. in Ward's sense) is not guaranteed

- **Escalability**: running time complexity is $\Omega(N^2)$; N = no. of objects

- **Interpretability**: an entire hierarchy (rather than a partition) is produced, which is the primary goal in many applications (e.g. taxonomy) and allows data visualisation, exploratory data analysis, etc.

- **Relational Computation**: Original data is not required, since the algorithms can operate just using a distance matrix

References

- Jain, A. K. and Dubes, R. C., Algorithms for Clustering Data, Prentice Hall, 1988
- Everitt, B. S., Landau, S., and Leese, M., Cluster Analysis, Arnold, 4th Edition, 2001
- Tan, P.-N., Steinbach, M., and Kumar, V., Introduction to Data Mining, Addison-Wesley, 2006
- Tan, P.-N., Steinbach, M., and Kumar, V., *Introduction to Data Mining*, Addison-Wesley, 2nd Edition, 2018
- Gan, G., Ma, C., and Wu, J., Data Clustering: Theory, Algorithms and Applications, ASA SIAM, 2007
- Gordon, A. D., “Hierarchical Classification”, Em Arabie et al. (Eds.), Clustering and Classification, World Scientific, 1996