CHAPTER 14 Hierarchical Clustering

Given n points in a d-dimensional space, the goal of hierarchical clustering is to create a sequence of nested partitions, which can be conveniently visualized via a tree or hierarchy of clusters, also called the cluster *dendrogram*. The clusters in the hierarchy range from the fine-grained to the coarse-grained – the lowest level of the tree (the leaves) consists of each point in its own cluster, whereas the highest level (the root) consists of all points in one cluster. Both of these may be considered to be *trivial* clusterings. At some intermediate level, we may find meaningful clusters. If the user supplies k, the desired number of clusters, we can choose the level at which there are k clusters.

There are two main algorithmic approaches to mine hierarchical clusters: agglomerative and divisive. Agglomerative strategies work in a bottom-up manner. That is, starting with each of the n points in a separate cluster, they repeatedly merge the most similar pair of clusters until all points are members of the same cluster. Divisive strategies do just the opposite, working in a top-down manner. Starting with all the points in the same cluster, they recursively split the clusters until all points are in separate clusters. In this chapter we focus on agglomerative strategies. We discuss some divisive strategies in Chapter 16, in the context of graph partitioning.

14.1 PRELIMINARIES

Given a dataset $\mathbf{D} = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$, where $\mathbf{x}_i \in \mathbb{R}^d$, a clustering $\mathcal{C} = \{C_1, \ldots, C_k\}$ is a partition of $\mathbf{D}$, that is, each cluster is a set of points $C_i \subseteq \mathbf{D}$, such that the clusters are pairwise disjoint $C_i \cap C_j = \emptyset$ (for all $i \neq j$), and $\cup_{i=1}^k C_i = \mathbf{D}$. A clustering $\mathcal{A} = \{A_1, \ldots, A_r\}$ is said to be nested in another clustering $\mathcal{B} = \{B_1, \ldots, B_s\}$ if and only if $r > s$, and for each cluster $A_i \in \mathcal{A}$, there exists a cluster $B_j \in \mathcal{B}$, such that $A_i \subseteq B_j$. Hierarchical clustering yields a sequence of n nested partitions $\mathcal{C}_1, \ldots, \mathcal{C}_n$, ranging from the trivial clustering $\mathcal{C}_1 = \big\{\{\mathbf{x}_1\}, \ldots, \{\mathbf{x}_n\}\big\}$ where each point is in a separate cluster, to the other trivial clustering $\mathcal{C}_n = \big\{\{\mathbf{x}_1, \ldots, \mathbf{x}_n\}\big\}$, where all points are in one cluster. In general, the clustering $\mathcal{C}_{t-1}$ is nested in the clustering $\mathcal{C}_t$. The cluster dendrogram is a rooted binary tree that captures this nesting structure, with edges between cluster $C_i \in \mathcal{C}_{t-1}$ and cluster $C_j \in \mathcal{C}_t$ if C_i is nested in C_j, that is, if $C_i \subset C_j$. In this way the dendrogram captures the entire sequence of nested clusterings.

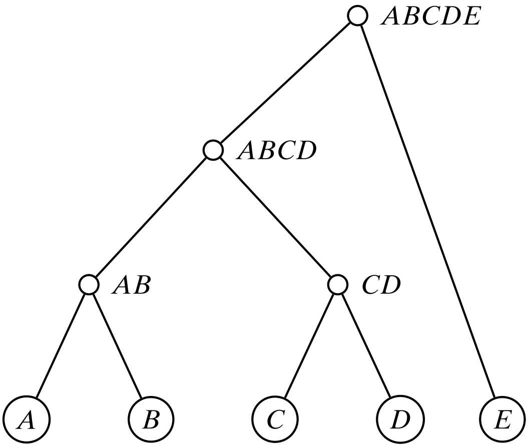


Figure 14.1. Hierarchical clustering dendrogram.

Example 14.1. Figure 14.1 shows an example of hierarchical clustering of five labeled points: A, B, C, D, and E. The dendrogram represents the following sequence of nested partitions:

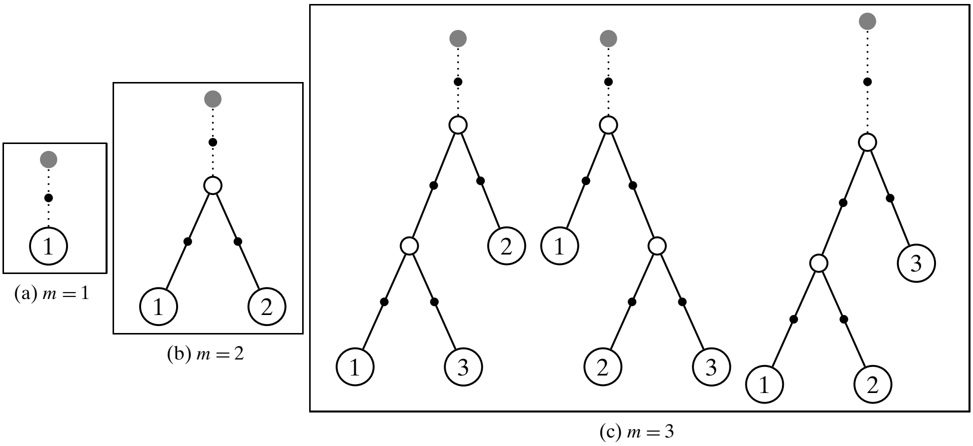
Clustering	Clusters
$\mathcal{C}_1$	$\{A\},\{B\},\{C\},\{D\},\{E\}$
$\mathcal{C}_2$	$\{AB\},\{C\},\{D\},\{E\}$
$\mathcal{C}_3$	$\{AB\},\{CD\},\{E\}$
$\mathcal{C}_4$	$\{ABCD\},\{E\}$
$\mathcal{C}_5$	$\{ABCDE\}$

with $\mathcal{C}_{t-1} \subset \mathcal{C}_t$ for $t = 2,\ldots,5$. We assume that A and B are merged before C and D.

Number of Hierarchical Clusterings

The number of different nested or hierarchical clusterings corresponds to the number of different binary rooted trees or dendrograms with n leaves with distinct labels. Any tree with t nodes has $t-1$ edges. Also, any rooted binary tree with m leaves has $m-1$ internal nodes. Thus, a dendrogram with m leaf nodes has a total of $t = m + m - 1 = 2m - 1$ nodes, and consequently $t - 1 = 2m - 2$ edges. To count the number of different dendrogram topologies, let us consider how we can extend a dendrogram with m leaves by adding an extra leaf, to yield a dendrogram with $m+1$ leaves. Note that we can add the extra leaf by splitting (i.e., branching from) any of the $2m-2$ edges. Further, we can also add the new leaf as a child of a new root, giving $2m - 2 + 1 = 2m - 1$ new dendrograms with $m+1$ leaves. The total number of different dendrograms with n leaves is thus obtained by the following product:

$$\prod_{m=1}^{n-1}(2m-1) = 1 \times 3 \times 5 \times 7 \times \cdots \times (2n-3) = (2n-3)!! \qquad (14.1)$$



(a) $m = 1$

(b) $m = 2$

(c) $m = 3$

Figure 14.2. Number of hierarchical clusterings.

The index m in Eq. (14.1) goes up to $n-1$ because the last term in the product denotes the number of dendrograms one obtains when we extend a dendrogram with $n-1$ leaves by adding one more leaf, to yield dendrograms with n leaves.

The number of possible hierarchical clusterings is thus given as $(2n-3)!!$, which grows extremely rapidly. It is obvious that a naive approach of enumerating all possible hierarchical clusterings is simply infeasible.

Example 14.2. Figure 14.2 shows the number of trees with one, two, and three leaves. The gray nodes are the virtual roots, and the black dots indicate locations where a new leaf can be added. There is only one tree possible with a single leaf, as shown in Figure 14.2a. It can be extended in only one way to yield the unique tree with two leaves in Figure 14.2b. However, this tree has three possible locations where the third leaf can be added. Each of these cases is shown in Figure 14.2c. We can further see that each of the trees with $m = 3$ leaves has five locations where the fourth leaf can be added, and so on, which confirms the equation for the number of hierarchical clusterings in Eq. (14.1).

14.2 AGGLOMERATIVE HIERARCHICAL CLUSTERING

In agglomerative hierarchical clustering, we begin with each of the n points in a separate cluster. We repeatedly merge the two closest clusters until all points are members of the same cluster, as shown in the pseudo-code given in Algorithm 14.1. Formally, given a set of clusters $\mathcal{C} = \{C_1, C_2, .., C_m\}$, we find the *closest* pair of clusters C_i and C_j and merge them into a new cluster $C_{ij} = C_i \cup C_j$. Next, we update the set of clusters by removing C_i and C_j and adding C_{ij}, as follows $\mathcal{C} = \big(\mathcal{C} \setminus \{C_i, C_j\}\big) \cup \{C_{ij}\}$. We repeat the process until $\mathcal{C}$ contains only one cluster. Because the number of clusters decreases by one in each step, this process results in a sequence of n nested clusterings. If specified, we can stop the merging process when there are exactly k clusters remaining.

ALGORITHM 14.1. Agglomerative Hierarchical Clustering Algorithm

```
AGGLOMERATIVECLUSTERING($\mathbf{D}, k$):
$\mathcal{C} \leftarrow \{C_i = \{\mathbf{x}_i\} \mid \mathbf{x}_i \in \mathbf{D}\}$ // Each point in separate cluster
$\boldsymbol{\Delta} \leftarrow \{\delta(\mathbf{x}_i, \mathbf{x}_j) \colon \mathbf{x}_i, \mathbf{x}_j \in \mathbf{D}\}$ // Compute distance matrix
repeat
    Find the closest pair of clusters $C_i, C_j \in \mathcal{C}$
    $C_{ij} \leftarrow C_i \cup C_j$ // Merge the clusters
    $\mathcal{C} \leftarrow \big(\mathcal{C} \setminus \{C_i, C_j\}\big) \cup \{C_{ij}\}$ // Update the clustering
    Update distance matrix $\boldsymbol{\Delta}$ to reflect new clustering
until $|\mathcal{C}| = k$
```

14.2.1 Distance between Clusters

The main step in the algorithm is to determine the closest pair of clusters. Several distance measures, such as single link, complete link, group average, and others discussed in the following paragraphs, can be used to compute the distance between any two clusters. The between-cluster distances are ultimately based on the distance between two points, which is typically computed using the Euclidean distance or L_2-*norm*, defined as

$$\delta(\mathbf{x}, \mathbf{y}) = \|\mathbf{x} - \mathbf{y}\|_2 = \left(\sum_{i=1}^{d} (x_i - y_i)^2\right)^{1/2}$$

However, one may use other distance metrics, or if available one may a user-specified distance matrix.

Single Link

Given two clusters C_i and C_j, the distance between them, denoted $\delta(C_i, C_j)$, is defined as the minimum distance between a point in C_i and a point in C_j

$$\delta(C_i, C_j) = \min\{\delta(\mathbf{x}, \mathbf{y}) \mid \mathbf{x} \in C_i, \mathbf{y} \in C_j\}$$

The name *single link* comes from the observation that if we choose the minimum distance between points in the two clusters and connect those points, then (typically) only a single link would exist between those clusters because all other pairs of points would be farther away.

Complete Link

The distance between two clusters is defined as the maximum distance between a point in C_i and a point in C_j:

$$\delta(C_i, C_j) = \max\{\delta(\mathbf{x}, \mathbf{y}) \mid \mathbf{x} \in C_i, \mathbf{y} \in C_j\}$$

The name *complete link* conveys the fact that if we connect all pairs of points from the two clusters with distance at most $\delta(C_i, C_j)$, then all possible pairs would be connected, that is, we get a complete linkage.

Group Average

The distance between two clusters is defined as the average pairwise distance between points in C_i and C_j:

$$\delta(C_i, C_j) = \frac{\sum_{\mathbf{x}\in C_i} \sum_{\mathbf{y}\in C_j} \delta(\mathbf{x}, \mathbf{y})}{n_i \cdot n_j}$$

where $n_i = |C_i|$ denotes the number of points in cluster C_i.

Mean Distance

The distance between two clusters is defined as the distance between the means or centroids of the two clusters:

$$\delta(C_i, C_j) = \delta(\boldsymbol{\mu}_i, \boldsymbol{\mu}_j) \tag{14.2}$$

where $\boldsymbol{\mu}_i = \frac{1}{n_i}\sum_{\mathbf{x}\in C_i} \mathbf{x}$.

Minimum Variance: Ward's Method

The distance between two clusters is defined as the increase in the sum of squared errors (SSE) when the two clusters are merged. The SSE for a given cluster C_i is given as

$$SSE_i = \sum_{\mathbf{x}\in C_i} \|\mathbf{x} - \boldsymbol{\mu}_i\|^2$$

which can also be written as

$$\begin{aligned} SSE_i &= \sum_{\mathbf{x}\in C_i} \|\mathbf{x} - \boldsymbol{\mu}_i\|^2 \\ &= \sum_{\mathbf{x}\in C_i} \mathbf{x}^T\mathbf{x} - 2\sum_{\mathbf{x}\in C_i} \mathbf{x}^T\boldsymbol{\mu}_i + \sum_{\mathbf{x}\in C_i} \boldsymbol{\mu}_i^T\boldsymbol{\mu}_i \\ &= \Big(\sum_{\mathbf{x}\in C_i} \mathbf{x}^T\mathbf{x}\Big) - n_i\boldsymbol{\mu}_i^T\boldsymbol{\mu}_i \end{aligned} \tag{14.3}$$

The SSE for a clustering $\mathcal{C} = \{C_1, \ldots, C_m\}$ is given as

$$SSE = \sum_{i=1}^{m} SSE_i = \sum_{i=1}^{m} \sum_{\mathbf{x}\in C_i} \|\mathbf{x} - \boldsymbol{\mu}_i\|^2$$

Ward's measure defines the distance between two clusters C_i and C_j as the net change in the SSE value when we merge C_i and C_j into C_{ij}, given as

$$\delta(C_i, C_j) = \Delta SSE_{ij} = SSE_{ij} - SSE_i - SSE_j \tag{14.4}$$

We can obtain a simpler expression for the Ward's measure by plugging Eq. (14.3) into Eq. (14.4), and noting that because $C_{ij} = C_i \cup C_j$ and $C_i \cap C_j = \emptyset$, we have $|C_{ij}| = n_{ij} = n_i + n_j$, and therefore

$$\begin{aligned}
\delta(C_i, C_j) &= \Delta SSE_{ij} \\
&= \sum_{\mathbf{z}\in C_{ij}} \left\|\mathbf{z} - \boldsymbol{\mu}_{ij}\right\|^2 - \sum_{\mathbf{x}\in C_i} \|\mathbf{x} - \boldsymbol{\mu}_i\|^2 - \sum_{\mathbf{y}\in C_j} \left\|\mathbf{y} - \boldsymbol{\mu}_j\right\|^2 \\
&= \sum_{\mathbf{z}\in C_{ij}} \mathbf{z}^T\mathbf{z} - n_{ij}\boldsymbol{\mu}_{ij}^T\boldsymbol{\mu}_{ij} - \sum_{\mathbf{x}\in C_i} \mathbf{x}^T\mathbf{x} + n_i\boldsymbol{\mu}_i^T\boldsymbol{\mu}_i - \sum_{\mathbf{y}\in C_j} \mathbf{y}^T\mathbf{y} + n_j\boldsymbol{\mu}_j^T\boldsymbol{\mu}_j \\
&= n_i\boldsymbol{\mu}_i^T\boldsymbol{\mu}_i + n_j\boldsymbol{\mu}_j^T\boldsymbol{\mu}_j - (n_i + n_j)\boldsymbol{\mu}_{ij}^T\boldsymbol{\mu}_{ij} \qquad (14.5)
\end{aligned}$$

The last step follows from the fact that $\sum_{\mathbf{z}\in C_{ij}} \mathbf{z}^T\mathbf{z} = \sum_{\mathbf{x}\in C_i} \mathbf{x}^T\mathbf{x} + \sum_{\mathbf{y}\in C_j} \mathbf{y}^T\mathbf{y}$. Noting that

$$\boldsymbol{\mu}_{ij} = \frac{n_i\boldsymbol{\mu}_i + n_j\boldsymbol{\mu}_j}{n_i + n_j}$$

we obtain

$$\boldsymbol{\mu}_{ij}^T\boldsymbol{\mu}_{ij} = \frac{1}{(n_i + n_j)^2}\left(n_i^2\boldsymbol{\mu}_i^T\boldsymbol{\mu}_i + 2n_in_j\boldsymbol{\mu}_i^T\boldsymbol{\mu}_j + n_j^2\boldsymbol{\mu}_j^T\boldsymbol{\mu}_j\right)$$

Plugging the above into Eq. (14.5), we finally obtain

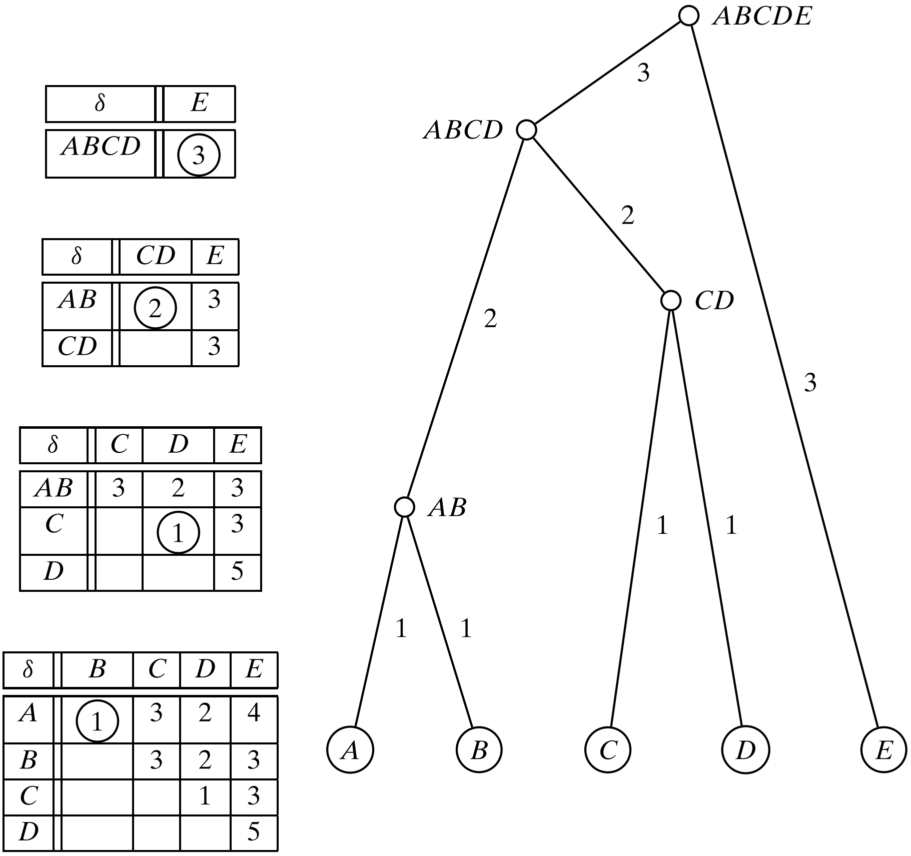
$$\begin{aligned}
\delta(C_i, C_j) &= \Delta SSE_{ij} \\
&= n_i\boldsymbol{\mu}_i^T\boldsymbol{\mu}_i + n_j\boldsymbol{\mu}_j^T\boldsymbol{\mu}_j - \frac{1}{(n_i + n_j)}\left(n_i^2\boldsymbol{\mu}_i^T\boldsymbol{\mu}_i + 2n_in_j\boldsymbol{\mu}_i^T\boldsymbol{\mu}_j + n_j^2\boldsymbol{\mu}_j^T\boldsymbol{\mu}_j\right) \\
&= \frac{n_i(n_i + n_j)\boldsymbol{\mu}_i^T\boldsymbol{\mu}_i + n_j(n_i + n_j)\boldsymbol{\mu}_j^T\boldsymbol{\mu}_j - n_i^2\boldsymbol{\mu}_i^T\boldsymbol{\mu}_i - 2n_in_j\boldsymbol{\mu}_i^T\boldsymbol{\mu}_j - n_j^2\boldsymbol{\mu}_j^T\boldsymbol{\mu}_j}{n_i + n_j} \\
&= \frac{n_in_j\left(\boldsymbol{\mu}_i^T\boldsymbol{\mu}_i - 2\boldsymbol{\mu}_i^T\boldsymbol{\mu}_j + \boldsymbol{\mu}_j^T\boldsymbol{\mu}_j\right)}{n_i + n_j} \\
&= \left(\frac{n_in_j}{n_i + n_j}\right)\left\|\boldsymbol{\mu}_i - \boldsymbol{\mu}_j\right\|^2
\end{aligned}$$

Ward's measure is therefore a weighted version of the mean distance measure because if we use Euclidean distance, the mean distance in Eq. (14.2) can be rewritten as

$$\delta(\boldsymbol{\mu}_i, \boldsymbol{\mu}_j) = \left\|\boldsymbol{\mu}_i - \boldsymbol{\mu}_j\right\|^2 \qquad (14.6)$$

We can see that the only difference is that Ward's measure weights the distance between the means by half of the harmonic mean of the cluster sizes, where the harmonic mean of two numbers n_1 and n_2 is given as $\frac{2}{\frac{1}{n_1}+\frac{1}{n_2}} = \frac{2n_1n_2}{n_1+n_2}$.

Example 14.3 (Single Link). Consider the single link clustering shown in Figure 14.3 on a dataset of five points, whose pairwise distances are also shown on the bottom left. Initially, all points are in their own cluster. The closest pair of points are (A, B) and (C, D), both with $\delta = 1$. We choose to first merge A and B, and derive a new distance matrix for the merged cluster. Essentially, we have to



δ	E
$ABCD$	3

δ	CD	E
AB	2	3
CD		3

δ	C	D	E
AB	3	2	3
C		1	3
D			5

δ	B	C	D	E
A	1	3	2	4
B		3	2	3
C			1	3
D				5

Figure 14.3. Single link agglomerative clustering.

compute the distances of the new cluster AB to all other clusters. For example, $\delta(AB, E) = 3$ because $\delta(AB, E) = \min\{\delta(A, E), \delta(B, E)\} = \min\{4, 3\} = 3$. In the next step we merge C and D because they are the closest clusters, and we obtain a new distance matrix for the resulting set of clusters. After this, AB and CD are merged, and finally, E is merged with $ABCD$. In the distance matrices, we have shown (circled) the minimum distance used at each iteration that results in a merging of the two closest pairs of clusters.

14.2.2 Updating Distance Matrix

Whenever two clusters C_i and C_j are merged into C_{ij}, we need to update the distance matrix by recomputing the distances from the newly created cluster C_{ij} to all other clusters C_r ($r \neq i$ and $r \neq j$). The Lance–Williams formula provides a general equation to recompute the distances for all of the cluster proximity measures we considered earlier; it is given as

$$\begin{aligned}\delta(C_{ij}, C_r) = \alpha_i \cdot \delta(C_i, C_r) + \alpha_j \cdot \delta(C_j, C_r) + \\ \beta \cdot \delta(C_i, C_j) + \gamma \cdot \big|\delta(C_i, C_r) - \delta(C_j, C_r)\big|\end{aligned} \tag{14.7}$$

Table 14.1. Lance–Williams formula for cluster proximity

Measure	α_i	α_j	β	γ
Single link	$\frac{1}{2}$	$\frac{1}{2}$	0	$-\frac{1}{2}$
Complete link	$\frac{1}{2}$	$\frac{1}{2}$	0	$\frac{1}{2}$
Group average	$\frac{n_i}{n_i+n_j}$	$\frac{n_j}{n_i+n_j}$	0	0
Mean distance	$\frac{n_i}{n_i+n_j}$	$\frac{n_j}{n_i+n_j}$	$\frac{-n_i \cdot n_j}{(n_i+n_j)^2}$	0
Ward's measure	$\frac{n_i+n_r}{n_i+n_j+n_r}$	$\frac{n_j+n_r}{n_i+n_j+n_r}$	$\frac{-n_r}{n_i+n_j+n_r}$	0

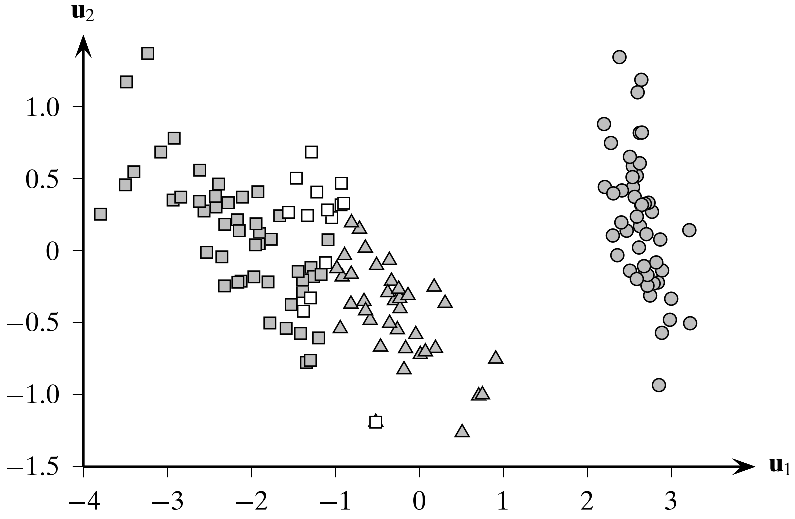


Figure 14.4. Iris dataset: complete link.

The coefficients $\alpha_i, \alpha_j, \beta$, and γ differ from one measure to another. Let $n_i = |C_i|$ denote the cardinality of cluster C_i; then the coefficients for the different distance measures are as shown in Table 14.1.

Example 14.4. Consider the two-dimensional Iris principal components dataset shown in Figure 14.4, which also illustrates the results of hierarchical clustering using the complete-link method, with $k = 3$ clusters. Table 14.2 shows the contingency table comparing the clustering results with the ground-truth Iris types (which are not used in clustering). We can observe that 15 points are misclustered in total; these points are shown in white in Figure 14.4. Whereas `iris-setosa` is well separated, the other two Iris types are harder to separate.

14.2.3 Computational Complexity

In agglomerative clustering, we need to compute the distance of each cluster to all other clusters, and at each step the number of clusters decreases by 1. Initially it takes

Table 14.2. Contingency table: clusters versus Iris types

	`iris-setosa`	`iris-virginica`	`iris-versicolor`
C_1 (circle)	50	0	0
C_2 (triangle)	0	1	36
C_3 (square)	0	49	14

$O(n^2)$ time to create the pairwise distance matrix, unless it is specified as an input to the algorithm.

At each merge step, the distances from the merged cluster to the other clusters have to be recomputed, whereas the distances between the other clusters remain the same. This means that in step t, we compute $O(n-t)$ distances. The other main operation is to find the closest pair in the distance matrix. For this we can keep the n^2 distances in a heap data structure, which allows us to find the minimum distance in $O(1)$ time; creating the heap takes $O(n^2)$ time. Deleting/updating distances from the heap takes $O(\log n)$ time for each operation, for a total time across all merge steps of $O(n^2 \log n)$. Thus, the computational complexity of hierarchical clustering is $O(n^2 \log n)$.

14.3 FURTHER READING

Hierarchical clustering has a long history, especially in taxonomy or classificatory systems, and phylogenetics; see, for example, Sokal and Sneath (1963). The generic Lance–Williams formula for distance updates appears in Lance and Williams (1967). Ward's measure is from Ward (1963). Efficient methods for single-link and complete-link measures with $O(n^2)$ complexity are given in Sibson (1973) and Defays (1977), respectively. For a good discussion of hierarchical clustering, and clustering in general, see Jain and Dubes (1988).

Defays, D. (Nov. 1977). "An efficient algorithm for a complete link method." *Computer Journal*, 20 (4): 364–366.

Jain, A. K. and Dubes, R. C. (1988). *Algorithms for Clustering Data*. Upper Saddle River, NJ: Prentice-Hall.

Lance, G. N. and Williams, W. T. (1967). "A general theory of classificatory sorting strategies 1. Hierarchical systems." *The Computer Journal*, 9 (4): 373–380.

Sibson, R. (1973). "SLINK: An optimally efficient algorithm for the single-link cluster method." *Computer Journal*, 16 (1): 30–34.

Sokal, R. R. and Sneath, P. H. (1963). *Principles of Numerical Taxonomy*. San Francisco: W.H. Freeman.

Ward, J. H. (1963). "Hierarchical grouping to optimize an objective function." *Journal of the American Statistical Association*, 58 (301): 236–244.

14.4 EXERCISES AND PROJECTS

Q1. Consider the 5-dimensional categorical data shown in Table 14.3.

Table 14.3. Data for Q1

Point	X_1	X_2	X_3	X_4	X_5
$\mathbf{x}_1$	1	0	1	1	0
$\mathbf{x}_2$	1	1	0	1	0
$\mathbf{x}_3$	0	0	1	1	0
$\mathbf{x}_4$	0	1	0	1	0
$\mathbf{x}_5$	1	0	1	0	1
$\mathbf{x}_6$	0	1	1	0	0

The similarity between categorical data points can be computed in terms of the number of matches and mismatches for the different attributes. Let n_{11} be the number of attributes on which two points $\mathbf{x}_i$ and $\mathbf{x}_j$ assume the value 1, and let n_{10} denote the number of attributes where $\mathbf{x}_i$ takes value 1, but $\mathbf{x}_j$ takes on the value of 0. Define n_{01} and n_{00} in a similar manner. The contingency table for measuring the similarity is then given as

		$\mathbf{x}_j$	
$\mathbf{x}_i$		1	0
	1	n_{11}	n_{10}
	0	n_{01}	n_{00}

Define the following similarity measures:

- Simple matching coefficient: $SMC(X_i, X_j) = \frac{n_{11}+n_{00}}{n_{11}+n_{10}+n_{01}+n_{00}}$
- Jaccard coefficient: $JC(X_i, X_j) = \frac{n_{11}}{n_{11}+n_{10}+n_{01}}$
- Rao's coefficient: $RC(X_i, X_j) = \frac{n_{11}}{n_{11}+n_{10}+n_{01}+n_{00}}$

Find the cluster dendrograms produced by the hierarchical clustering algorithm under the following scenarios:

(a) We use single link with RC.

(b) We use complete link with SMC.

(c) We use group average with JC.

Q2. Given the dataset in Figure 14.5, show the dendrogram resulting from the single-link hierarchical agglomerative clustering approach using the L_1-*norm* as the distance between points

$$\delta(\mathbf{x}, \mathbf{y}) = \sum_{a=1}^{2} |x_{ia} - y_{ia}|$$

Whenever there is a choice, merge the cluster that has the lexicographically smallest labeled point. Show the cluster merge order in the tree, stopping when you have $k = 4$ clusters. Show the full distance matrix at each step.

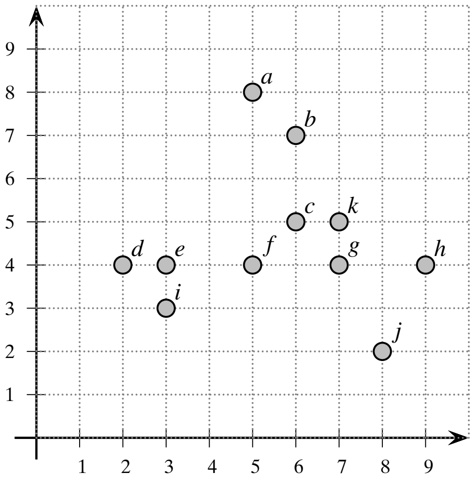


Figure 14.5. Dataset for Q2.

Table 14.4. Dataset for Q3

	A	B	C	D	E
A	0	1	3	2	4
B		0	3	2	3
C			0	1	3
D				0	5
E					0

Q3. Using the distance matrix from Table 14.4, use the average link method to generate hierarchical clusters. Show the merging distance thresholds.

Q4. Prove that in the Lance–Williams formula [Eq. (14.7)]

(a) If $\alpha_i = \frac{n_i}{n_i+n_j}$, $\alpha_j = \frac{n_j}{n_i+n_j}$, $\beta = 0$ and $\gamma = 0$, then we obtain the group average measure.

(b) If $\alpha_i = \frac{n_i+n_r}{n_i+n_j+n_r}$, $\alpha_j = \frac{n_j+n_r}{n_i+n_j+n_r}$, $\beta = \frac{-n_r}{n_i+n_j+n_r}$ and $\gamma = 0$, then we obtain Ward's measure.

Q5. If we treat each point as a vertex, and add edges between two nodes with distance less than some threshold value, then the single-link method corresponds to a well known graph algorithm. Describe this graph-based algorithm to hierarchically cluster the nodes via single-link measure, using successively higher distance thresholds.