Cluster analysis or clustering is the assignment of a set of observations into subsets (called clusters) so that observations in the same cluster are similar in some sense.

Hierarchical Clustering

(most common approach)

To decide which clusters should be combined a measure of dissimilarity between sets of observations is required.

This is achieved by use of an appropriate metric (a measure of distance between pairs of observations), and a linkage criteria that determines how clustered are formed based the distance between objects.
Hierarchical Tree
(like a directory structure on your computer)

**Agglomerative**

- Related files grouped in Directories
- Individual files

Hierarchical Clustering
(conceptual view)

**Step 1:**
Make every point its own cluster

**Step 2:**
Find most similar pair of clusters and merge into parent cluster

Hierarchical Clustering
(conceptual view)

**Step 1:**
Make every point its own cluster

**Step 2:**
Find most similar pair of clusters and merge into parent cluster

**Step 3:**
Repeat

Hierarchical Clustering
(conceptual view)
Hierarchical Clustering
(conceptual view)

**Step 1:**
Make every point its own cluster

**Step 2:**
Find most similar pair of clusters and merge into parent cluster

**Step 3:**
Repeat

**Step 4:**
Work up to next level

---

Matlab implementation of H. Clustering

1. Find the similarity or dissimilarity between every pair of objects in the data set. In this step, you calculate the distance between objects using the pdist function.

2. Group the objects into a cluster tree. In this step, you link pairs of objects that are in close proximity using the linkage function. The linkage function uses the distance information generated in step 1.

3. Determine where to cut the hierarchical tree into clusters. In this step, you use the cluster function to prune branches off the bottom of the hierarchical tree, and assign all the objects below each cut to a single cluster.
y = pdist(X) computes the Euclidean distance between pairs of objects in n-by-p data matrix X.

Rows of X correspond to observations or samples; columns correspond to variables that have been measured (p total).

\[ X = \begin{pmatrix} X_{11}, X_{12}, \ldots, X_{1p} \\ X_{21}, X_{22}, \ldots, X_{2p} \end{pmatrix} \]

y is a row vector of length n(n-1)/2, corresponding to pairs of observations in X.

Y(1) gives the distance between the data in row 1 and row 2

\[ Y(1) = \sqrt{ (X_{11} - X_{21})^2 + (X_{12} - X_{22})^2 + \ldots + (X_{1p} - X_{2p})^2 } \]

Parameter 1 Parameter 2

For Euclidean

Many other measures of distance can be used by pdist.

<table>
<thead>
<tr>
<th>Measure</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>'euclidean'</td>
<td>Euclidean distance function</td>
</tr>
<tr>
<td>'minkowski'</td>
<td>Minkowski distance function</td>
</tr>
<tr>
<td>'cityblock'</td>
<td>City block distance function</td>
</tr>
</tbody>
</table>

Some of the more common distance measures

'seuclidean'
Standardized Euclidean distance. Each coordinate difference between rows in X is scaled by dividing by the corresponding element of the standard deviation S=nanstd(X). To specify another value for S, use D=pdist(X,'seuclidean',S). Useful if the parameters have different units.

'cityblock' or Manhattan

\[ Y(1) = | X_{11} - X_{21} | + | X_{12} - X_{22} | + \ldots + | X_{1p} - X_{2p} | \]

In a city you must walk perpendicular blocks rather than crossing diagonally (like Euclidian)
'correlation'
One minus the sample correlation between objects.

\[ 1 - r_{x1x2} \]

Consider two sample X1 and X2 with p parameters measured for each

\[ X_1 = \{ x_{11}, x_{12}, \ldots, x_{1p} \} \]

\[ X_2 = \{ x_{21}, x_{22}, \ldots, x_{2p} \} \]

\[ r_{12} = \frac{\sum_{i=1}^{p} (x_{1i} - \bar{x}_1)(x_{2i} - \bar{x}_2)}{pS_{x1}S_{x2}} \]

X1 - parameters
X2 - parameters

More correlated they are the closer they are

Z is a (m-1)-by-3 matrix, where m is the number of observations in the original data. Columns 1 and 2 of Z contain cluster indices linked in pairs to form a binary tree. The 3rd column is the distance between clusters (or the link length)

Example

Individual points are clusters 1 .... 7

\[ X = [3 \ 1.7; \ 1 \ 1; \ 2 \ 2.5; \ 2 \ 3; \ 1.2 \ 1.1 \ 1.5; \ 3 \ 1] \]

\[ Y = \text{pdist}(X); \]

\[ Z = \text{linkage}(Y) \]

\[ Z = \begin{bmatrix}
2.0000 & 5.0000 & 0.2000 \\
3.0000 & 4.0000 & 0.5000 \\
8.0000 & 6.0000 & 0.5099 \\
1.0000 & 7.0000 & 0.7000 \\
11.0000 & 9.0000 & 1.2806 \\
12.0000 & 10.0000 & 1.3454 \\
\end{bmatrix} \]

This says ....

Cluster 2 is a distance 0.2 from cluster 5:

\[ \text{sqrt}((1-1.2)^2 + (1-1)^2) = 0.2 \]

Z = linkage(y) creates an hierarchical cluster tree. The input y is a Euclidean distance matrix formatted as a vector, as returned by pdist.

Z = linkage(y,method) creates the tree using the specified method. Methods differ from one another in how they measure the distance between clusters. Available methods are listed in the following table.

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>'average'</td>
<td>Unweighted average distance (UPGMA).</td>
</tr>
<tr>
<td>'centroid'</td>
<td>Centroid distance (UPGMC). y must contain Euclidean distances.</td>
</tr>
<tr>
<td>'complete'</td>
<td>Furthest distance.</td>
</tr>
<tr>
<td>'median'</td>
<td>Weighted center of mass distance (WPGMC). y must contain Euclidean distances.</td>
</tr>
<tr>
<td>'single'</td>
<td>Shortest distance. This is the default.</td>
</tr>
<tr>
<td>'ward'</td>
<td>Inter squared distance (minimum variance algorithm). y must contain Euclidean distances.</td>
</tr>
<tr>
<td>'unweighted'</td>
<td>Weighted average distance (WPGMA).</td>
</tr>
</tbody>
</table>

Z = linkage(x) returns a condensed distance matrix Z. Z(i,j) is the distance between clusters i and j.

Y is the pdist output

\[ Y = \begin{bmatrix}
2.1100 & 1.6401 & 1.2806 & 1.9313 & 1.9105 & 0.7200 \\
2.2361 & 1.8028 & 0.5099 & 2.0000 & 2.1541 & 1.7493 \\
1.3454 & 1.8028 & 0.5099 & 1.8000 & 1.9647 & 0.7200 \\
\end{bmatrix} \]
$Z = \text{linkage}(Y)$

What about cluster 8, where does it come from?

Cluster eight is formed by merging cluster 2 and cluster 5, the closest pair.

The closest distance between 6 and the new 8 is 0.5099

$T = \text{cluster}(Z, \text{maxclust}, n)$ constructs a maximum of $n$ clusters using the 'distance' criterion. cluster finds the smallest height at which a horizontal cut through the tree leaves $n$ or fewer clusters.

$\text{c} = \text{cluster}(Z, \text{maxclust}, 3)$

Objects 2, 5 and 6 form cluster 3
Objects 1 and 7 form cluster 2
Objects 3 and 4 form cluster 1

$Z = \text{linkage}(Y)$

$Z = \begin{array}{ccc}
2.0000 & 5.0000 & 0.2000 \\
3.0000 & 4.0000 & 0.5000 \\
6.0000 & 8.0000 & 0.5099 \\
1.0000 & 7.0000 & 0.7000 \\
9.0000 & 11.0000 & 1.2866 \\
10.0000 & 12.0000 & 1.3454 \\
\end{array}$

$T = \text{cluster}(Z, \text{maxclust}, 3)$

$\text{c} = \text{cluster}(Z, \text{cutoff}, 0.8, \text{criterion}, \text{distance})$

$c2 = \begin{array}{c}
3 \\
2 \\
1 \\
1 \\
3 \\
3 \\
\end{array}$

$\text{c} = \text{cluster}(Z, \text{cutoff}, 0.4, \text{criterion}, \text{distance})$

$c2 = \begin{array}{c}
3 \\
5 \\
1 \\
4 \\
2 \\
6 \\
\end{array}$
Example using Iris data (150 x 4)

- Load iris data: `load fisheriris`
- Calculate distance: `d = pdist(meas)`
- Calculate linkage: `z = linkage(d)`
- Plot dendrogram: `dendrogram(z, 0)`

% the 0 tells dendrogram to display all lengths

```matlab
c = cluster(z,'maxclust',2);
A = find(c==1) % tells me which flowers in cluster 1
B = find(c==2) % tells me which flowers in cluster 2
```

- 100 flowers in B
- 50 flowers in A