**Hierarchical clustering algorithm**

**Hierarchical clustering:**

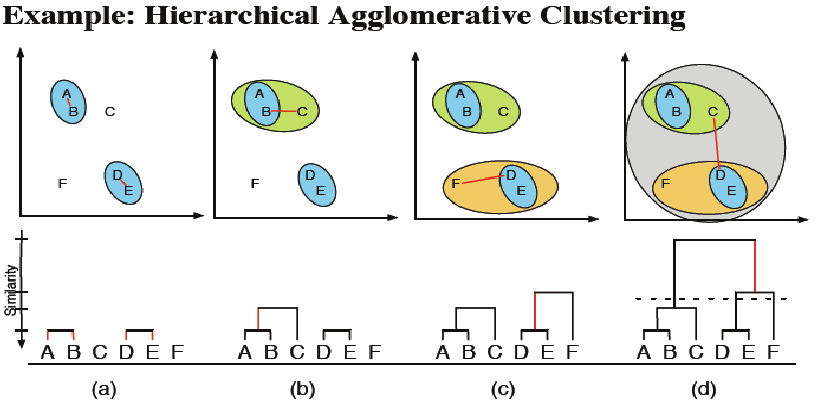
It is an unsupervised learning technique. Two sequential clustering strategies for constructing a tree of clusters. The hierarchical clustering algorithm consists of two different flavours: agglomerative clustering and divisive clustering. Both of these techniques involve building some type of dendrogram or tree that reveals the relationships between the data objects in a data set (see the image above).

Agglomerative clustering begins with each data object being its own cluster. The next step is to identify which clusters are the closest to others using a distance metric. The similar objects are then formed into bigger clusters of objects.

Each iteration of this approach creates bigger and bigger clusters at each level until one giant cluster of all the objects remains. Each level of clusters can then be analyzed to determine the groups of similar data objects that formed.

Divisive clustering is just the opposite of hierarchical in that it starts with one giant cluster containing all data objects, and then divides into smaller clusters after each iteration. It ends when each data object is its own cluster.

**Ex:**



The Hierarchical clustering essentially are of two types:

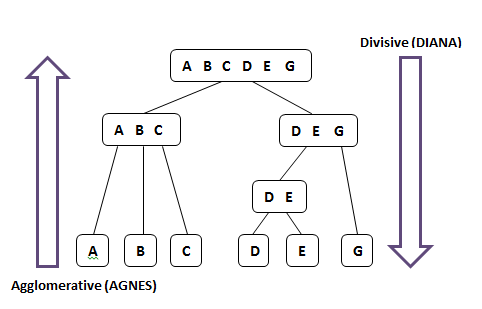
1. **Agglomerative**:
2. **Divisive**

**Agglomerative**:

1. This is a "bottom up" approach
2. Initially each observation starts in its own cluster.
3. Then merge these atomic clusters into larger and larger clusters.

**Divisive**:

1. This is a "top down" approach:
2. all observations start in one cluster
3. Then the clusters subdivided into smaller and smaller clusters.



Agglomerative clustering:

|  |  |  |
| --- | --- | --- |
|  | x | y |
| P1 | 0.40 | 0.53 |
| P2 | 0.22 | 0.38 |
| P3 | 0.35 | 0.32 |
| P4 | 0.26 | 0.19 |
| P5 | 0.08 | 0.41 |
| P6 | 0.45 | 0.30 |

y

0.6

0.5

0.4

0.3

0.2

0.1

0 0.1 0.2 0.3 0.4 0.5 x

Calculate Euclidean distance create the distance matrix:

**Euclidean distance [(x,y),(a,b)] = sqrt ((x-a)^2 + (x-b)^2)**

Distance (p1,p2) = sqrt ((0.40-0.22)^2 + (0.53-0.38)^2) = 0.23

Distance (p1,p3) = sqrt ((0.40-0.35)^2 + (0.53-0.32)^2) = 0.22

Distance (p1,p4) = sqrt ((0.40-0.26)^2 + (0.53-0.19)^2) = 0.37

Distance (p1,p5) = sqrt ((0.40-0.08)^2 + (0.53-0.41)^2) = 0.34

Distance (p1,p6) = sqrt ((0.40-0.45)^2 + (0.53-0.30)^2) = 0.23

Distance (p2,p3) = sqrt ((0.22-0.35)^2 + (0.38-0.32)^2) = 0.15

Distance (p2,p4) = sqrt ((0.22-0.26)^2 + (0.38-0.19)^2) = 0.20

Distance (p2, p5) = sqrt ((0.22-0.08)^2 + (0.38-0.41)^2) = 0.14

Distance (p2,p6) = sqrt ((0.22-0.45)^2 + (0.38-0.30)^2) = 0.15

Distance (p3,p4) = sqrt ((0.35-0.26)^2 + (0.32-0.19)^2) = 0.15

Distance (p3,p5) = sqrt ((0.35-0.08)^2 + (0.32-0.41)^2) = 0.28

Distance (p3,p6) = sqrt ((0.35-0.45)^2 + (0.32-0.30)^2) = 0.11

Distance (p4,p5) = sqrt ((0.26-0.08)^2 + (0.19-0.41)^2) = 0.29

Distance (p4,p6) = sqrt ((0.26-0.45)^2 + (0.19-0.3.0)^2) = 0.22

Distance (p5,p6) = sqrt ((0.08-0.45)^2 + (0.41-0.30)^2) = 0.39

**The distance matrix is**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **P1** | **P2** | **P3** | **P4** | **P5** | **P6** |
| **P1** | **0** |  |  |  |  |  |
| **P2** | **0.23** | **0** |  |  |  |  |
| **P3** | **0.22** | **0.15** | **0** |  |  |  |
| **P4** | **0.37** | **0.20** | **0.15** | **0** |  |  |
| **P5** | **0.34** | **0.14** | **0.28** | **0.29** | **0** |  |
| **P6** | **0.23** | **0.25** | **0.11** | **0.22** | **0.39** | **0** |

**To find the minimum distance:**

P3, p6 is minimum distance =0.11

y

0.6

0.5

0.4

0.3

0.2

0.1

0 0.1 0.2 0.3 0.4 0.5 x

Dendogram:

p3 p6

**Recalculate the distance matrix**

1. Min[dist(p3,p6),p1]

Min [dist (p3, p1), (p6, p1)]

[0.22, 0.23]

Min (0.22)

1. Min[dist(p3,p6),p2]

Min [dist (p3, p2), (p6, p2)]

[0.15, 0.25]

Min (0.15)

1. Min [dist (p3, p6), p4]

Min [dist (p3, p4), (p6, p4)]

[0.15, 0.22]

Min (0.15)

1. Min [dist (p3, p6), p5]

Min [dist (p3, p5), (p6, p5)]

[0.28, 0.39]

Min (0.28)

**Find the minimum distance:**

p2, p5 minimum distance is 0.14

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **P1** | **P2** | **P3,p6** | **P4** | **P5** |
| **P1** | **0** |  |  |  |  |
| **P2** | **0.23** | **0** |  |  |  |
| **P3,p6** | **0.22** | **0.15** | **0** |  |  |
| **P4** | **0.37** | **0.20** | **0.15** | **0** |  |
| **P5** | **0.34** | **0.14** | **0.28** | **0.29** | **0** |

Dendogram:

p3, p6 p2, p5

**The distance matrix for cluster p2, p5:**

To update the distance matrix

1. Min[dist(p2,p5),p1)

Min [dist (p2, p1), (p5, p1)]

Min [0.23, 0.34]

Min (0.23)

1. Min[dist(p2,p5),(p3,p6)]

Min [dist (p2, (p3, p6), (p5, (p3, p6)]

Min [0.15, 0.28]

Min (0.15)

1. Min[dist(p2,p5),p4)

Min [dist (p2, p4), (p5, p4)]

Min [0.20, 0.29]

Min (0.20)

**Find the minimum distance:**

(p3, p6), (p2, p5) minimum distance is 0.15

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **P1** | **P2,p5** | **P3,p6** | **P4** |
| **P1** | **0** |  |  |  |
| **P2,p5** | **0.23** | **0** |  |  |
| **P3,p6** | **0.22** | **0.15** | **0** |  |
| **P4** | **0.37** | **0.20** | **0.15** | **0** |

Dendogram:

p3, p6, p2, p5,

**The distance matrix for clusters p3, p6, p2, p5:**

To update the distance matrix

1. Min[dist(p2,p5),(p3,p6),p1)

Min [dist ((p2, p5), p1), ((p3, p6), p1)]

Min [0.23, 0.22]

Min (0.22)

1. Min[dist(p2,p5),(p3,p6),p4]

Min [dist ((p2, p5), p4)), ((p3, p6), p4))]

Min [0.20, 0.15]

Min (0.15)

To update the distance matrix for clusters p2, p5 and p3, p6

|  |  |  |  |
| --- | --- | --- | --- |
|  | **p1** | **p2,p5,P3,p6** | **p4** |
| **p1** | **0** |  |  |
| **p2,p5,p3,p6** | **0.22** | **0** |  |
| **p4** | **0.37** | **0.15** | **0** |

Dendogram:

p3, p6, p2, p5, p4

To update the distance matrix

1. Min[dist(p2,p5,p3,p6),p4,p1)

Min [dist ((p2, p5, p3, p6), p1), (p4, p1)]

Min [0.22, 0.37]

Min (0.22)

To update the distance matrix for clusters p2, p5, p3, p6, p4

|  |  |  |
| --- | --- | --- |
|  | **p1** | **p2,p5,P3,p6,p4** |
| **p1** | **0** |  |
| **p2,p5,p3,p6** | **0.22** | **0** |

Dendogram:

p3, p6, p2, p5, p4, p1