Hierarchical Clustering

- Produces a set of nested clusters organized as a hierarchical tree
- Can be visualized as a dendrogram
  - A tree like diagram that records the sequences of merges or splits
Strengths of Hierarchical Clustering

- Do not have to assume any particular number of clusters
  - Any desired number of clusters can be obtained by ‘cutting’ the dendogram at the proper level

- They may correspond to meaningful taxonomies
  - Example in biological sciences (e.g., animal kingdom, phylogeny reconstruction, …)
Hierarchical Clustering

- Two main types of hierarchical clustering
  - Agglomerative:
    - Start with the points as individual clusters
    - At each step, merge the closest pair of clusters until only one cluster (or k clusters) left
  - Divisive:
    - Start with one, all-inclusive cluster
    - At each step, split a cluster until each cluster contains a point (or there are k clusters)

- Traditional hierarchical algorithms use a similarity or distance matrix
  - Merge or split one cluster at a time
Agglomerative Clustering Algorithm

- More popular hierarchical clustering technique
- Basic algorithm is straightforward
  - Compute the proximity matrix
  - Let each data point be a cluster
  - Repeat
    - Merge the two closest clusters
    - Update the proximity matrix
  - Until only a single cluster remains

- Key operation is the computation of the proximity of two clusters
  - Different approaches to defining the distance between clusters distinguish the different algorithms
Starting Situation

- Start with clusters of individual points and a proximity matrix

```
   p1 p2 p3 p4 p5 ... 
  p1 . . . . . . . 
  p2 . . . . . . . 
  p3 . . . . . . . 
  p4 . . . . . . . 
  p5 . . . . . . . 
          . . . . . 
          . . . . . 
Proximity Matrix
```
Intermediate Situation

After some merging steps, we have some clusters

<table>
<thead>
<tr>
<th></th>
<th>C1</th>
<th>C2</th>
<th>C3</th>
<th>C4</th>
<th>C5</th>
</tr>
</thead>
<tbody>
<tr>
<td>C1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Proximity Matrix
Intermediate Situation

- We want to merge the two closest clusters (C2 and C5) and update the proximity matrix.

![Proximity Matrix]

<table>
<thead>
<tr>
<th></th>
<th>C1</th>
<th>C2</th>
<th>C3</th>
<th>C4</th>
<th>C5</th>
</tr>
</thead>
<tbody>
<tr>
<td>C1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
After Merging

The question is “How do we update the proximity matrix?”

Proximity Matrix

<table>
<thead>
<tr>
<th></th>
<th>C1</th>
<th>C5</th>
<th>C3</th>
<th>C4</th>
</tr>
</thead>
<tbody>
<tr>
<td>C1</td>
<td>?</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C2 U C5</td>
<td>?</td>
<td>?</td>
<td>?</td>
<td>?</td>
</tr>
<tr>
<td>C3</td>
<td>?</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C4</td>
<td>?</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
How to Define Inter-Cluster Similarity

- MIN
- MAX
- Group Average
- Distance Between Centroids
- Other methods driven by an objective function
  - Ward’s Method uses squared error

Proximity Matrix

```
<table>
<thead>
<tr>
<th></th>
<th>p1</th>
<th>p2</th>
<th>p3</th>
<th>p4</th>
<th>p5</th>
<th>...</th>
</tr>
</thead>
<tbody>
<tr>
<td>p1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>p2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>p3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>p4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>p5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
```
How to Define Inter-Cluster Similarity

- \textbf{MIN}
- \textbf{MAX}
- Group Average
- Distance Between Centroids
- Other methods driven by an objective function
  - Ward’s Method uses squared error

\begin{table}[h]
\centering
\begin{tabular}{|c|c|c|c|c|}
\hline
 & p1 & p2 & p3 & p4 & p5 \\
\hline
p1 & & & & & \\
\hline
p2 & & & & & \\
\hline
p3 & & & & & \\
\hline
p4 & & & & & \\
\hline
p5 & & & & & \\
\hline
\end{tabular}
\caption{Proximity Matrix}
\end{table}
How to Define Inter-Cluster Similarity

- MIN
- MAX
- Group Average
- Distance Between Centroids
- Other methods driven by an objective function
  - Ward’s Method uses squared error
How to Define Inter-Cluster Similarity

- MIN
- MAX
- Group Average
- Distance Between Centroids
- Other methods driven by an objective function
  - Ward’s Method uses squared error

Proximity Matrix
How to Define Inter-Cluster Similarity

- MIN
- MAX
- Group Average
- Distance Between Centroids
- Other methods driven by an objective function
  - Ward’s Method uses squared error
Cluster Similarity: MIN or Single Link

- Similarity of two clusters is based on the two most similar (closest) points in the different clusters
  - Determined by one pair of points, i.e., by one link in the proximity graph.

<table>
<thead>
<tr>
<th></th>
<th>I1</th>
<th>I2</th>
<th>I3</th>
<th>I4</th>
<th>I5</th>
</tr>
</thead>
<tbody>
<tr>
<td>I1</td>
<td>1,00</td>
<td>0,90</td>
<td>0,10</td>
<td>0,65</td>
<td>0,20</td>
</tr>
<tr>
<td>I2</td>
<td>0,90</td>
<td>1,00</td>
<td>0,70</td>
<td>0,60</td>
<td>0,50</td>
</tr>
<tr>
<td>I3</td>
<td>0,10</td>
<td>0,70</td>
<td>1,00</td>
<td>0,40</td>
<td>0,30</td>
</tr>
<tr>
<td>I4</td>
<td>0,65</td>
<td>0,60</td>
<td>0,40</td>
<td>1,00</td>
<td>0,80</td>
</tr>
<tr>
<td>I5</td>
<td>0,20</td>
<td>0,50</td>
<td>0,30</td>
<td>0,80</td>
<td>1,00</td>
</tr>
</tbody>
</table>
Hierarchical Clustering: MIN

Nested Clusters

Dendrogram
Strength of MIN

- Can handle non-elliptical shapes
Limitations of MIN

- Sensitive to noise and outliers
  - Particular case: chain effect
Cluster Similarity: MAX or Complete Linkage

- Similarity of two clusters is based on the two least similar (most distant) points in the different clusters
  - Determined by all pairs of points in the two clusters

<table>
<thead>
<tr>
<th></th>
<th>I1</th>
<th>I2</th>
<th>I3</th>
<th>I4</th>
<th>I5</th>
</tr>
</thead>
<tbody>
<tr>
<td>I1</td>
<td>1.00</td>
<td>0.90</td>
<td>0.10</td>
<td>0.65</td>
<td>0.20</td>
</tr>
<tr>
<td>I2</td>
<td>0.90</td>
<td>1.00</td>
<td>0.70</td>
<td>0.60</td>
<td>0.50</td>
</tr>
<tr>
<td>I3</td>
<td>0.10</td>
<td>0.70</td>
<td>1.00</td>
<td>0.40</td>
<td>0.30</td>
</tr>
<tr>
<td>I4</td>
<td>0.65</td>
<td>0.60</td>
<td>0.40</td>
<td>1.00</td>
<td>0.80</td>
</tr>
<tr>
<td>I5</td>
<td>0.20</td>
<td>0.50</td>
<td>0.30</td>
<td>0.80</td>
<td>1.00</td>
</tr>
</tbody>
</table>
Hierarchical Clustering: MAX

Nested Clusters

Dendrogram
Strength of MAX

- Less susceptible to noise and outliers
Limitations of MAX

- Tends to break large clusters
- Biased towards globular clusters
Cluster Similarity: Group Average

- Proximity of two clusters is the average of pairwise proximity between points in the two clusters.

\[
proximity(C_i, C_j) = \frac{\sum_{x \in C_i} \sum_{y \in C_j} \text{proximity}(x, y)}{m_i \times m_j}
\]

- Need to use average connectivity for scalability since total proximity favors large clusters

```
<table>
<thead>
<tr>
<th></th>
<th>I1</th>
<th>I2</th>
<th>I3</th>
<th>I4</th>
<th>I5</th>
</tr>
</thead>
<tbody>
<tr>
<td>I1</td>
<td>1,00</td>
<td>0,90</td>
<td>0,10</td>
<td>0,65</td>
<td>0,20</td>
</tr>
<tr>
<td>I2</td>
<td>0,90</td>
<td>1,00</td>
<td>0,70</td>
<td>0,60</td>
<td>0,50</td>
</tr>
<tr>
<td>I3</td>
<td>0,10</td>
<td>0,70</td>
<td>1,00</td>
<td>0,40</td>
<td>0,30</td>
</tr>
<tr>
<td>I4</td>
<td>0,65</td>
<td>0,60</td>
<td>0,40</td>
<td>1,00</td>
<td>0,80</td>
</tr>
<tr>
<td>I5</td>
<td>0,20</td>
<td>0,50</td>
<td>0,30</td>
<td>0,80</td>
<td>1,00</td>
</tr>
</tbody>
</table>
```
Hierarchical Clustering: Group Average

Nested Clusters

Dendrogram
Hierarchical Clustering: Group Average

- Compromise between Single and Complete Link

- Strengths
  - Less susceptible to noise and outliers

- Limitations
  - Biased towards globular clusters
Cluster Similarity: Ward’s Method

- Similarity of two clusters is based on the increase in squared error when two clusters are merged
  - Similar to group average if distance between points is distance squared

- Less susceptible to noise and outliers

- Biased towards globular clusters

- Hierarchical analogue of K-means
  - Can be used to initialize K-means
Hierarchical Clustering: Comparison

- **Group Average**
- **Ward’s Method**
- **MIN**
- **MAX**
Hierarchical Clustering: Time and Space requirements

- $O(N^2)$ space since it uses the proximity matrix.
  - $N$ is the number of points.

- $O(N^3)$ time in many cases
  - There are $N$ steps and at each step the proximity matrix (size: $O(N^2)$) must be updated and searched
  - Complexity can be reduced to $O(N^2 \log(N))$ time for some approaches
Hierarchical Clustering: Problems and Limitations

- Once a decision is made to combine two clusters, it cannot be undone

- No objective function is directly minimized

- Different schemes have problems with one or more of the following:
  - Sensitivity to noise and outliers
  - Difficulty handling different sized clusters and convex shapes
  - Breaking large clusters
Single-link HAC vs Graphs

Cut edges with $D > 0.12$

Clusters = connected components
Single-link HAC vs Graphs

Cut edges with $D > 0.15$

1

2

3

4

5

6

Cut edges with $D > 0.18$

1

2

3

4

5

6

© Tan, Steinbach, Kumar
Introduction to Data Mining

4/18/2004 77
Complete-link HAC vs Graphs

Clusters ~ cliques – constraint: earlier cliques have precedence

Cut edges with D>0.15

Cut edges with D>0.25

© Tan, Steinbach, Kumar
Introduction to Data Mining

4/18/2004
**MST: Divisive Hierarchical Clustering**

- Build MST (Minimum Spanning Tree)
  - Start with a tree that consists of any point
  - In successive steps, look for the closest pair of points \((p, q)\) such that one point \((p)\) is in the current tree but the other \((q)\) is not
  - Add \(q\) to the tree and put an edge between \(p\) and \(q\)
MST: Divisive Hierarchical Clustering

Algorithm 7.5 MST Divisive Hierarchical Clustering Algorithm

1: Compute a minimum spanning tree for the proximity graph.
2: repeat
3: Create a new cluster by breaking the link corresponding to the largest distance (smallest similarity).
4: until Only singleton clusters remain