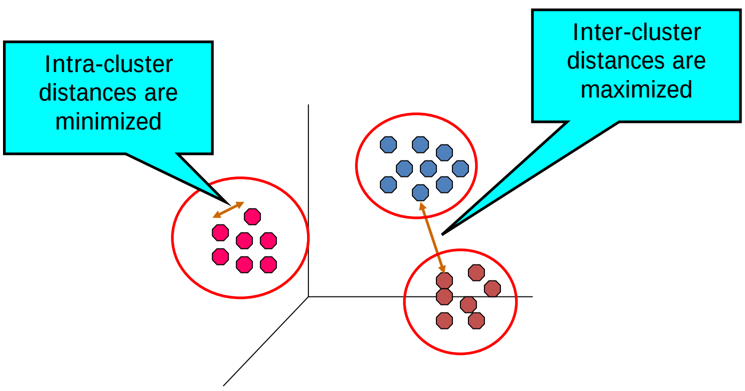
UNIVERSITAT DE BARCELONA

Hierarchical clustering

Prof. Miquel Salicrú
Prof. Sergi Civit

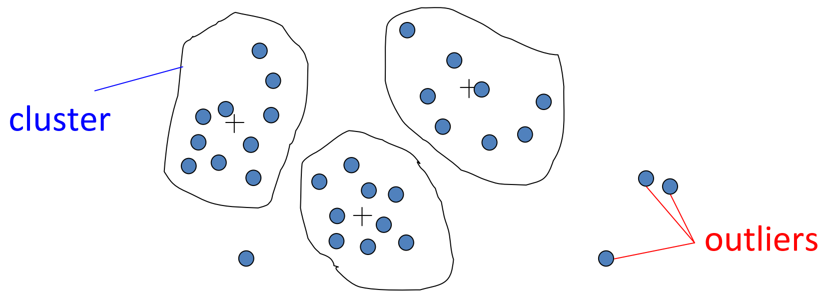
What is clustering?

A **grouping** of data objects such that the objects **within a group are similar** (or related) to one another **and different from (or unrelated to) the objects in other groups**



Outliers

- **Outliers** are **objects that do not belong to any cluster** or form clusters of very small cardinality



- In some applications we are interested in discovering outliers, not clusters (outlier analysis)

Why do we cluster?

- Clustering : given a collection of data objects group them so that
  - Similar to one another within the same cluster
  - Dissimilar to the objects in other clusters
- Clustering results are used
  - As a stand-alone tool to get insight into data distribution
    - Visualization of clusters may unveil important information
  - As a preprocessing step for other algorithms
    - Efficient indexing or compression often relies on clustering
    - Group representation in 2D or 3D graphic

Applications of clustering?

- **Marketing**
  - Identify patterns of behavior
- **Image Processing**
  - Cluster images based on their visual content
- **Web**
  - Cluster groups of users based on their access patterns on webpages
  - Cluster webpages based on their content
- **Bioinformatics**
  - Cluster similar proteins together (similarity with respect to chemical structure and/or functionality etc)

Basic questions

- How many groups?
- What is a good partition of the objects?
- How many methods we have to perform Cluster analysis?
  - What clustering procedure? (hierarchical clustering methods (agglomerative vs divisive) nonhierarchical clustering methods)
  - Aims and scope
  - What is the best method?
- Clustering results: “real groups” or artifact?
  - How to evaluate the performance of clustering algorithms?
  - Artificial groups?

The clustering task

Objective	Group structure or "cluster" (according to distances between populations)
Method	Hierarchical: agglomerative/divisive Not hierarchical
Information (loss)	Correlation coefficient: Pearson, Spearman Stress
Interpretation	What is a good partition? How many groups? Groups. Make sense?
Conclusion	Answer the question How do these results fit in with results from other studies?

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)

Agglomerative method: basic algorithm

1. Compute the distance matrix between the input data points
2. Let each data point be a cluster
3. **Repeat**
4. Merge the two closest clusters
5. Update the distance matrix
6. **Until** only a single cluster remains

Remark. Key operation is the computation of the distance between two clusters: different definitions of the distance between clusters lead to different algorithms

Algorithm-recursive process 1/2

Starting point:

- $\{P_1, P_2, \dots, P_s\}$ be the set (objects, populations) to classify
- $d_{ij}=d(P_i, P_j)$ distance between objects

Step1

- Identify the objects P_{io}, P_{jo} at minimum distance

$$l_1 = d(P_{io}, P_{jo}) = \min_{i,j}\{d(P_i, P_j)\}$$

- Create the "clustering" at $\psi(l_1)=\{P_1, P_2, \dots, P_{io} \cup P_{jo}, \dots, P_s\}$ level
- Define the distance $d^{(1)}$ at the classification level $\psi(l_1)$

$$d^{(1)}(P_i, P_j) = d^{(o)}(P_i, P_j) \quad \text{si} \quad \{i, j\} \cap \{i0, j0\} = 0$$

$$d^{(1)}(P_{io} \cup P_{jo}, P_k) = f(d(P_{io}, P_{jo}), d(P_{io}, P_k), d(P_{jo}, P_k))$$

Algorithm-recursive process 2/2

Step n

- Starting point: $\psi(l_{n-1})=\{A_1, A_2,\ldots,\ldots, A_{s-n+1}\}$, $d^{(n-1)}(A_i, A_j)$
- Identify the objects A_{io}, A_{jo} nearest

$$l_n = d^{(n-1)}(A_{io}, A_{jo}) = \min_{i,j}\{d^{(n-1)}(A_i, A_j)\}$$

- Create the "clustering" at $\psi(l_n)=\{A_1,\ldots, A_{io}\cup A_{jo},\ldots, A_{s-n+1}\}$ level
- Define the distance $d^{(n)}$ in the partition level $\psi(l_n)$

$$d^{(n)}(A_i, A_j) = d^{(n-1)}(A_i, A_j) \text{ si } \{i, j\} \cap \{i0, j0\} = 0$$

$$d^{(n)}(A_{io} \cup A_{jo}, A_k) = f(d^{(n-1)}(A_{io}, A_{jo}), d^{(n-1)}(A_{io}, A_k), d^{(n-1)}(A_{jo}, A_k))$$

Ultrametric Distance

$$d_u(P_i, P_j) = \min\{l_\alpha / P_i, P_j \in A_h \subset \psi(l_\alpha)\}$$

Distance between Clusters 1/2

- **Single link**: smallest distance between an element in one cluster and an element in the other, i.e.,

  $$dis(K_i, K_j) = \min(t_{ip}, t_{jq})$$

- **Complete link**: largest distance between an element in one cluster and an element in the other, i.e.,

  $$dis(K_i, K_j) = \max(t_{ip}, t_{jq})$$

- **Average**: avg distance between an element in one cluster and an element in the other, i.e.,

  $$dis(K_i, K_j) = avg(t_{ip}, t_{jq})$$

Distance between Clusters 2/2

- **Centroid**: distance between the centroids of two clusters, i.e.,

$$dis(K_i, K_j) = dis(C_i, C_j)$$

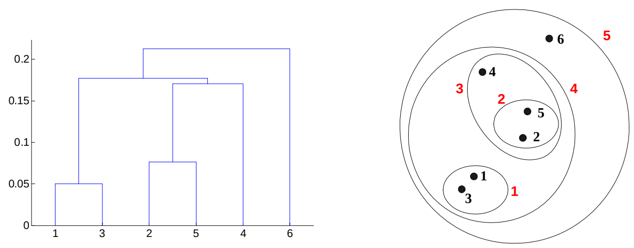
- **Medoid**: distance between the medoids of two clusters, i.e.,

$$dis(K_i, K_j) = dis(M_i, M_j)$$

Medoid: one chosen, centrally located object in the cluster

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

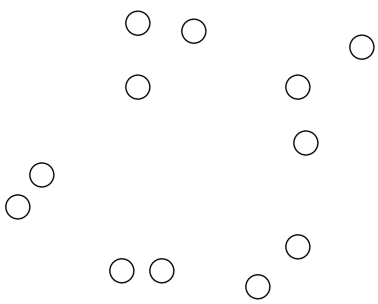


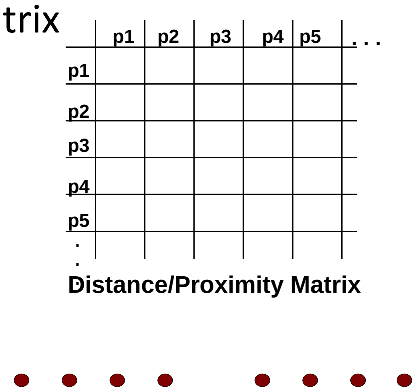
Strengths of Hierarchical Clustering

- No assumptions on the number of clusters
  - Any desired number of clusters can be obtained by 'cutting' the dendogram at the proper level
- Hierarchical clusterings may correspond to meaningful taxonomies
  - In biological sciences (e.g., phylogeny reconstruction, etc)
  - web (e.g., product catalogs)
  - etc

Input/ Initial setting

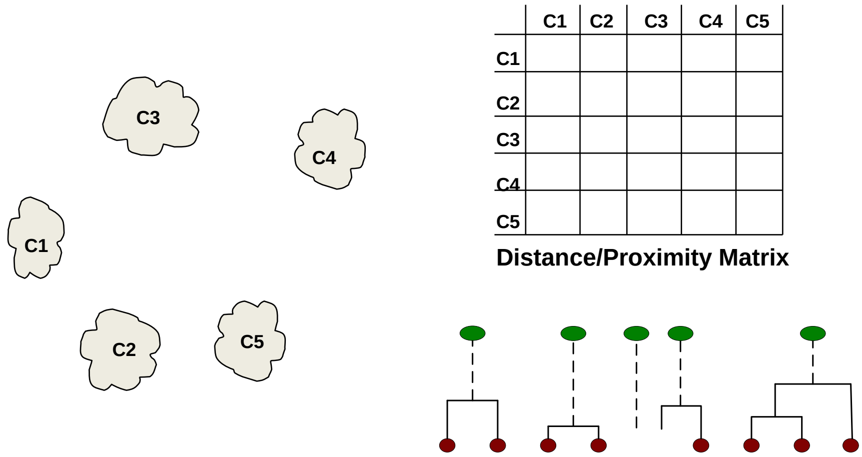
- Start with clusters of individual points and a distance/proximity matrix





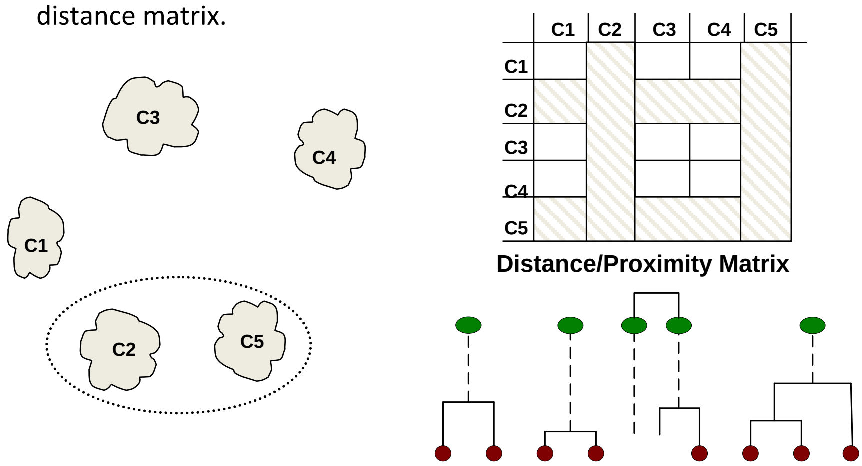
Intermediate State

- After some merging steps, we have some clusters

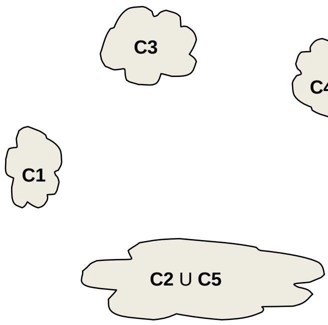


Intermediate State

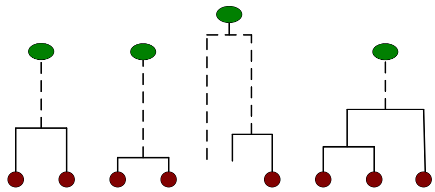
- Merge the two closest clusters (C2 and C5) and update the distance matrix.



After Merging



	C1	C2 ∪ C5	C3	C4
C1		?		
C2 ∪ C5	?	?	?	?
C3		?		
C4		?		



Distance between two clusters

- Each cluster is a set of points
- How do we define distance between two sets of points
  - Lots of alternatives
  - Not an easy task

Distance between two clusters

- **Single-link distance** between clusters C_i and C_j is the ***minimum distance*** between any object in C_i and any object in C_j

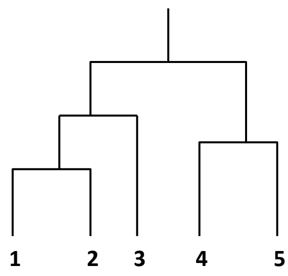
- The distance is **defined by the two most similar objects**

$$D_{sl}(C_i, C_j) = \min_{x,y}\{d(x,y) \mid x \in C_i, y \in C_j\}$$

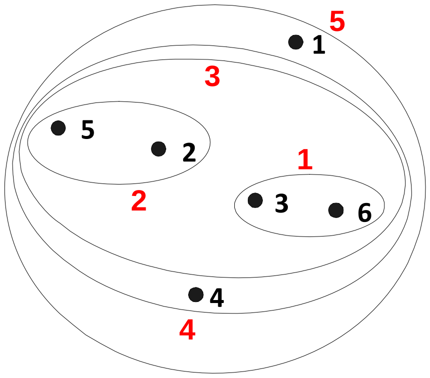
Single-link clustering: example

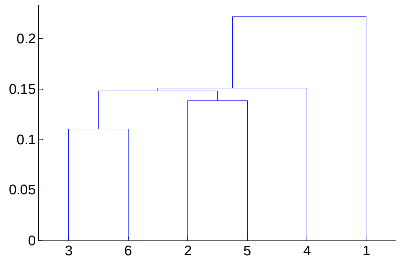
- Determined by one pair of points, i.e., by one link in the proximity graph.

	I1	I2	I3	I4	I5
I1	1.00	0.90	0.10	0.65	0.20
I2	0.90	1.00	0.70	0.60	0.50
I3	0.10	0.70	1.00	0.40	0.30
I4	0.65	0.60	0.40	1.00	0.80
I5	0.20	0.50	0.30	0.80	1.00



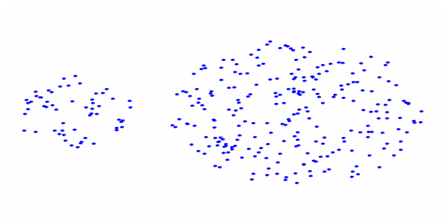
Single-link clustering: example



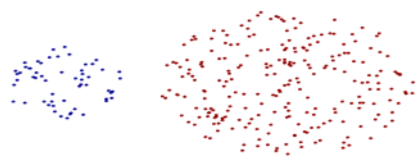


Nested Clusters **Dendrogram**

Strengths of single-link clustering



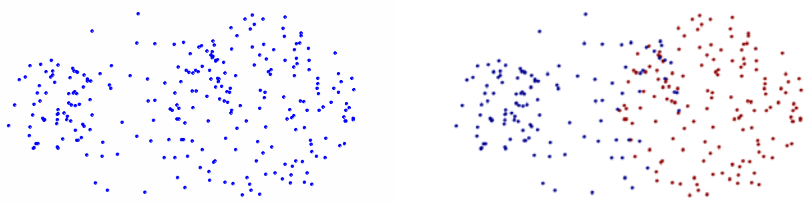
Original Points



Two Clusters

• **Can handle non-elliptical shapes**

Limitations of single-link clustering



Original Points   Two Clusters

• Sensitive to noise and outliers
• It produces long, elongated clusters

Distance between two clusters

- **Complete-link distance** between clusters C_i and C_j is the ***maximum distance*** between any object in C_i and any object in C_j

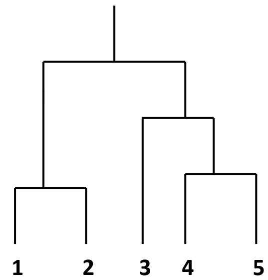
- The distance is **defined by the two most dissimilar objects**

$$D_{cl}(C_i, C_j) = \max_{x,y}\{d(x,y) \mid x \in C_i, y \in C_j\}$$

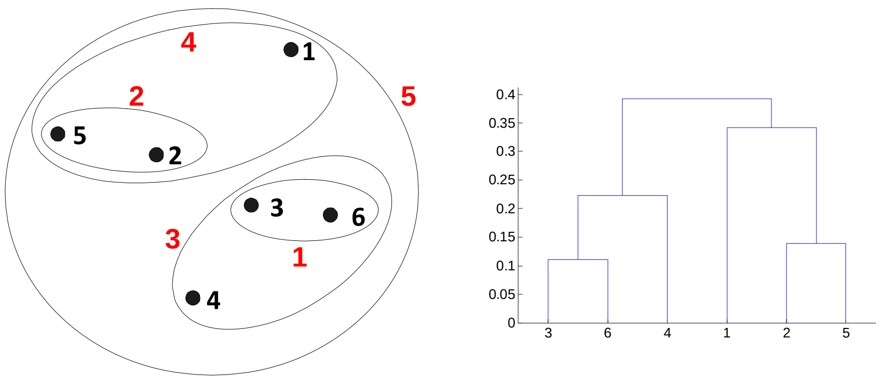
Complete-link clustering: example

- Distance between clusters is determined by the two most distant points in the different clusters

	I1	I2	I3	I4	I5
I1	1.00	0.90	0.10	0.65	0.20
I2	0.90	1.00	0.70	0.60	0.50
I3	0.10	0.70	1.00	0.40	0.30
I4	0.65	0.60	0.40	1.00	0.80
I5	0.20	0.50	0.30	0.80	1.00



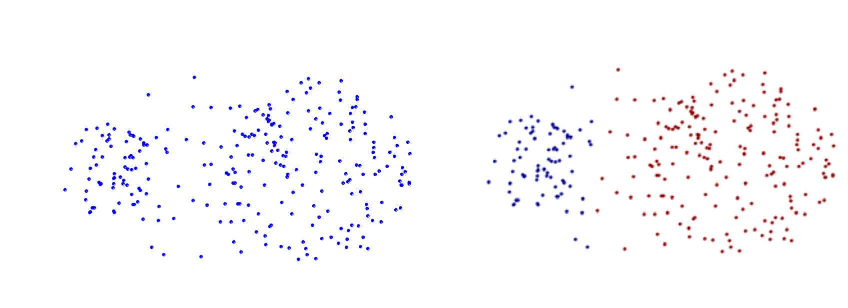
Complete-link clustering: example



Nested Clusters

Dendrogram

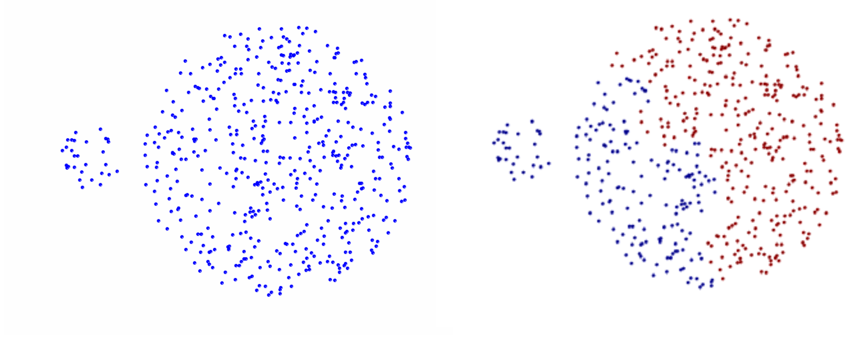
Strengths of complete-link clustering



Original Points Two Clusters

- More balanced clusters (with equal diameter)
- Less susceptible to noise

Limitations of complete-link clustering



Original Points Two Clusters

- Tends to break large clusters
- All clusters tend to have the same diameter – small clusters are merged with larger ones

Distance between two clusters

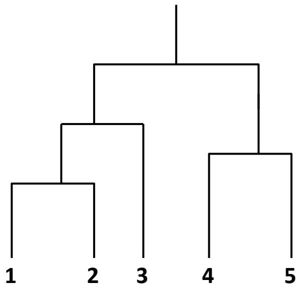
- **Group average distance** between clusters C_i and C_j is the ***average distance*** between any object in C_i and any object in C_j

$$D_{avg}(C_i, C_j) = \frac{1}{|C_i| \times |C_j|} \sum_{x \in C_i, y \in C_j} d(x, y)$$

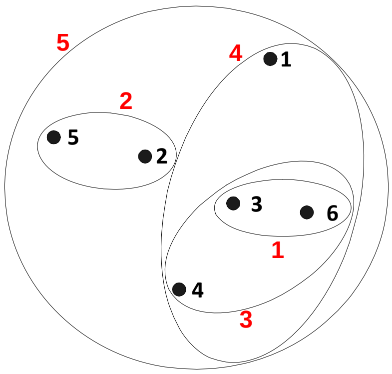
Average-link clustering: example

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

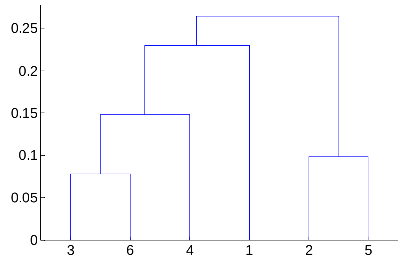
	I1	I2	I3	I4	I5
I1	1.00	0.90	0.10	0.65	0.20
I2	0.90	1.00	0.70	0.60	0.50
I3	0.10	0.70	1.00	0.40	0.30
I4	0.65	0.60	0.40	1.00	0.80
I5	0.20	0.50	0.30	0.80	1.00



Average-link clustering: example



Nested Clusters



Dendrogram

Average-link clustering: discussion

- Compromise between Single and Complete Link
- Strengths
  - Less susceptible to noise and outliers
- Limitations
  - Biased towards globular clusters

Distance between two clusters

- **Centroid distance** between clusters **C_i** and **C_j** is the distance between the centroid **r_i** of **C_i** and the centroid **r_j** of **C_j**

$$D_{centroids}(C_i, C_j) = d(r_i, r_j)$$

Distance between two clusters

- **Ward's distance** between clusters **C_i** and **C_j** is the ***difference*** between the ***total within cluster sum of squares for the two clusters separately***, and the ***within cluster sum of squares resulting from merging the two clusters*** in cluster **C_{ij}**

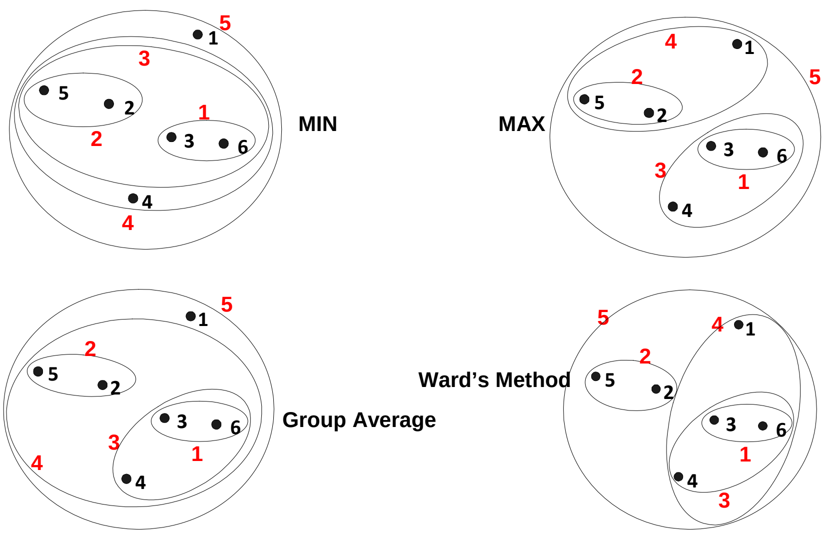
$$D_w(C_i, C_j) = \sum_{x \in C_{ij}} (x - r_{ij})^2 - \left(\sum_{x \in C_i} (x - r_i)^2 + \sum_{x \in C_j} (x - r_j)^2\right)$$

- **r_i**: centroid of **C_i**
- **r_j**: centroid of **C_j**
- **r_{ij}**: centroid of **C_{ij}**

Ward's distance for clusters

- Similar to group average and centroid distance
- 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



Divisive hierarchical clustering

- Start with a single cluster composed of all data points
- Split this into components
- Continue recursively
- *Monothetic* divisive methods split clusters using one variable/dimension at a time
- *Polythetic* divisive methods make splits on the basis of all variables together
- Any intercluster distance measure can be used
- Computationally intensive, less widely used than agglomerative methods

Model-based clustering

- Assume data generated from **k** probability distributions
- ***Goal:*** find the distribution parameters
- ***Algorithm:*** Expectation Maximization (EM)
- ***Output:*** Distribution parameters and a **soft** assignment of points to clusters