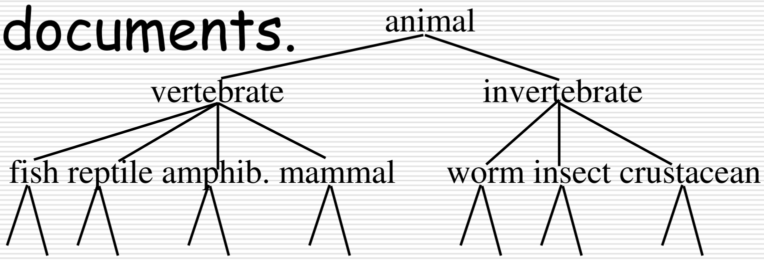
Hierarchical Clustering

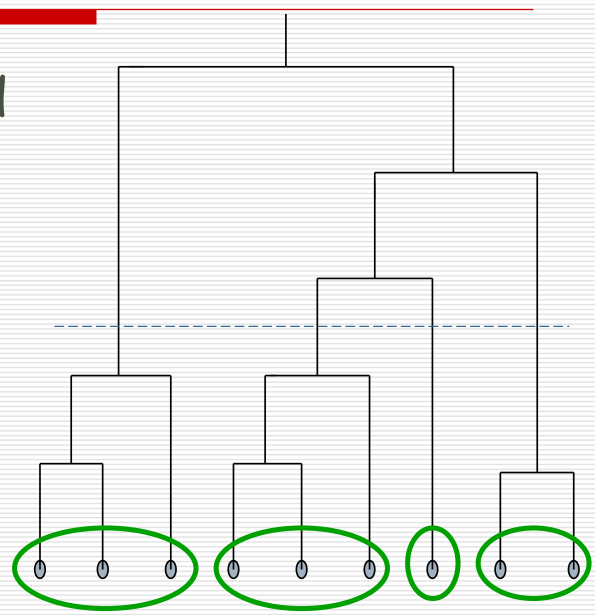
- Build a tree-based hierarchical taxonomy (*dendrogram*) from a set of documents.



- One approach: recursive application of a partitional clustering algorithm.

Dendogram: Hierarchical Clustering

- Clustering obtained by cutting the dendrogram at a desired level: each connected component forms a cluster.



The dendogram

- The y-axis of the dendogram represents the combination similarities, i.e. the similarities of the clusters merged by a the horizontal lines for a particular y
- Assumption: The merge operation is monotonic, i.e. if $s_1,..,s_{k-1}$ are successive combination similarities, then $s_1 \geq s_2 \geq ... \geq s_{k-1}$ must hold

Hierarchical Agglomerative Clustering (HAC)

- Starts with each doc in a separate cluster
  - then repeatedly joins the closest pair of clusters, until there is only one cluster.
- The history of merging forms a binary tree or hierarchy.

Closest pair of clusters

- Many variants to defining closest pair of clusters
- Single-link
  - Similarity of the *most* cosine-similar (single-link)
- Complete-link
  - Similarity of the "furthest" points, the *least* cosine-similar
- Centroid
  - Clusters whose centroids (centers of gravity) are the most cosine-similar
- Average-link
  - Average cosine between pairs of elements

Single Link Agglomerative Clustering

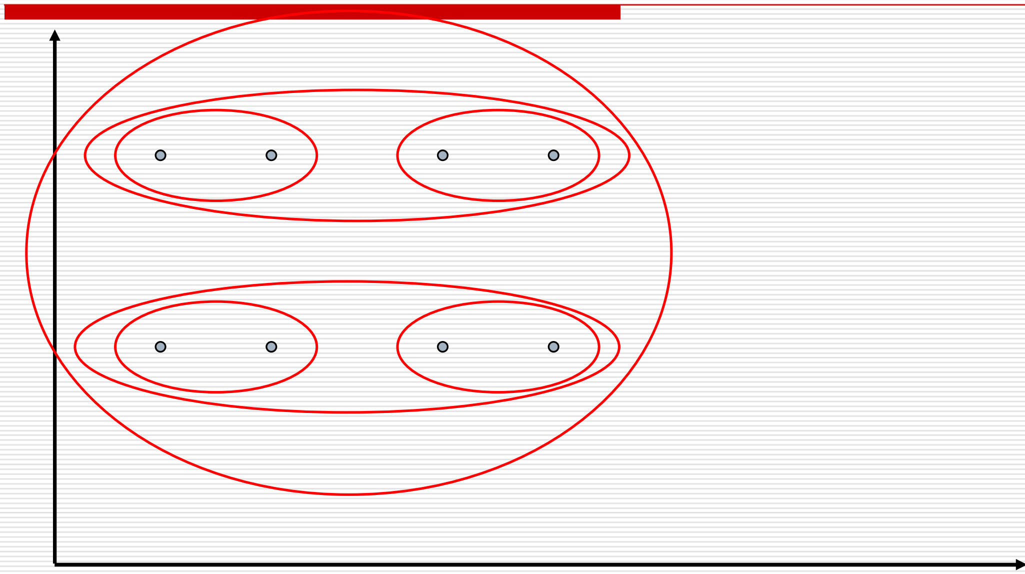
- Use maximum similarity of pairs:

$$sim(c_i, c_j) = \max_{x \in c_i, y \in c_j} sim(x, y)$$

- Can result in "straggly" (long and thin) clusters due to chaining effect.
- After merging c_i and c_j, the similarity of the resulting cluster to another cluster, c_k, is:

$$sim((c_i \cup c_j), c_k) = \max(sim(c_i, c_k), sim(c_j, c_k))$$

Single Link Example



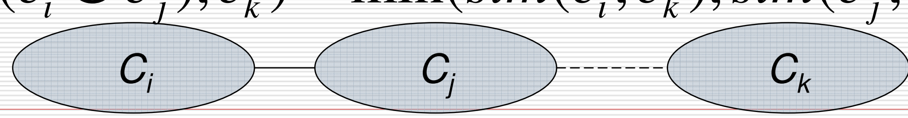
Complete Link Agglomerative Clustering

- Use minimum similarity of pairs:

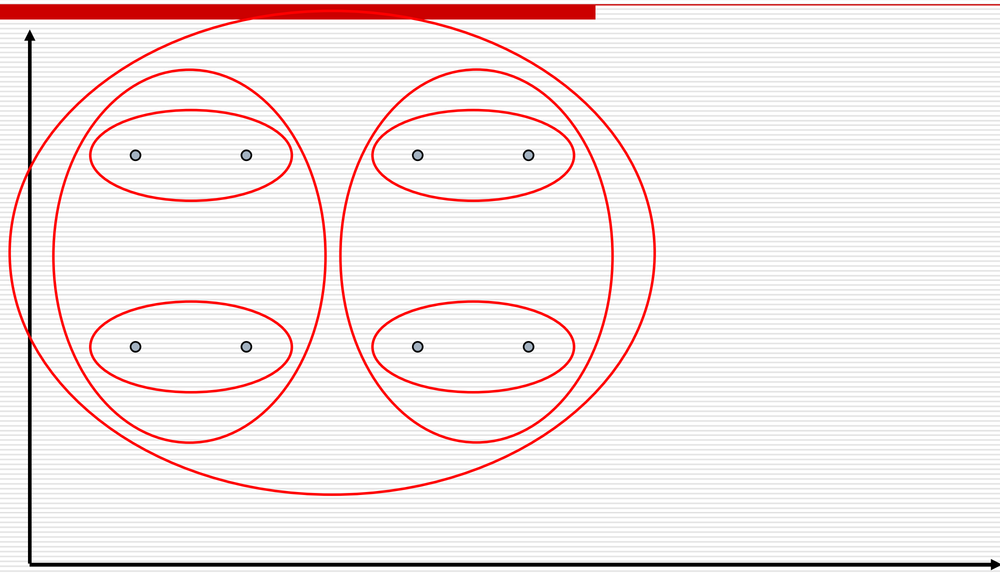
$$sim(c_i, c_j) = \min_{x \in c_i, y \in c_j} sim(x, y)$$

- Makes "tighter," spherical clusters that are typically preferable.
- After merging c_i and c_j, the similarity of the resulting cluster to another cluster, c_k, is:

$$sim((c_i \cup c_j), c_k) = \min(sim(c_i, c_k), sim(c_j, c_k))$$



Complete Link Example



Graph theoretical interpretation

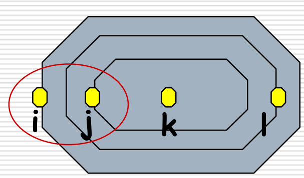
- Single-link Clustering as connected component of a graph
  - If $G(\Delta_k)$ is the graph that links all data points with a distance of at most Δ_k, then the clusters are the connected components of $G(\Delta_k)$
- Complete-link as cliques of a graph
  - If $G(\Delta_k)$ is the graph that links all data points with a distance of at most Δ_k, then the clusters are the cliques of $G(\Delta_k)$
- This motivates the terms single-link and complete-link clustering

Computational Complexity

- In the first iteration, all HAC methods need to compute similarity of all pairs of n individual instances which is $O(n^2)$.
- In each of the subsequent n–2 merging iterations, compute the distance between the most recently created cluster and all other existing clusters.
- In order to maintain an overall $O(n^2)$ performance, computing similarity to each other cluster must be done in constant time.
  - Else $O(n^2 \log n)$ or $O(n^3)$ if done naively

Best-Merge Persistency

- The single-link agglomerative clustering is best-merge persistent
- Suppose that the best merge cluster for k is j
- Then, after merging j with a third cluster i ≠ k, the merger of i and j will be the k's best merge cluster
- As a consequence, we can keep the best merge candidates for the merged cluster as one of the two best merge candidates for the merged clusters



Single-link and Complete-link drawbacks

- Single link clustering can produce straggling clusters. Since the merge criterion is local, it can cause the chaining effect
- Complete-link clustering pays to much attention to outliers, i.e. points that do not fit well in the global structure of the clusters

Group Average Agglomerative Clustering

- Similarity of two clusters = average similarity of all pairs within merged cluster.

$$sim_{ga}(\omega_i,\omega_j) = \frac{\sum_{d_k \in \omega_i \cup \omega_j} \sum_{d_l \in \omega_i \cup \omega_j, d_l \neq d_k} d_k d_l}{(N_i+N_j)(N_i+N_j-1)}$$

- Compromise between single and complete link.
- An alternative to group-average clustering is centroid clustering

$$sim_{cent}(\omega_i,\omega_j) = \frac{1}{N_i N_j} \sum_{d_k \in \omega_i} \sum_{d_l \in \omega_j, d_l \neq d_k} d_k d_l$$

Computing Group Average and Centroid similarities

- Always maintain sum of vectors in each cluster.

$$s(\omega_j) = \sum_{d_k \in \omega_j} d_k$$

- Compute similarity of clusters in constant time:

$$sim_{ga}(\omega_i, \omega_j) = \frac{s^2(\omega_i \cup \omega_j) - (N_i + N_j)}{(N_i + N_j)(N_i + N_j - 1)}$$

$$sim_{cent}(\omega_i, \omega_j) = \frac{s(\omega_i) s(\omega_j)}{N_i N_j}$$

Summarizing

Single-link	Max sim of any two points	$O(N^2)$	Chaining effect
Complete-link	Min sim of any two points	$O(N^2 \log N)$	Sensitive to outliers
Centroid	Similarity of centroids	$O(N^2 \log N)$	Non monotonic
Group-average	Avg sim of any two points	$O(N^2 \log N)$	OK