Data Mining and Machine Learning: Fundamental Concepts and Algorithms

dataminingbook.info

Mohammed J. Zaki[1] Wagner Meira Jr.[2]

[1]Department of Computer Science
Rensselaer Polytechnic Institute, Troy, NY, USA

[2]Department of Computer Science
Universidade Federal de Minas Gerais, Belo Horizonte, Brazil

Chapter 14: Hierarchical Clustering

Hierarchical Clustering

The goal of hierarchical clustering is to create a sequence of nested partitions, which can be conveniently visualized via a tree or hierarchy of clusters, also called the cluster *dendrogram*.

The clusters in the hierarchy range from the fine-grained to the coarse-grained – the lowest level of the tree (the leaves) consists of each point in its own cluster, whereas the highest level (the root) consists of all points in one cluster.

Agglomerative hierarchical clustering methods work in a bottom-up manner. Starting with each of the n points in a separate cluster, they repeatedly merge the most similar pair of clusters until all points are members of the same cluster.

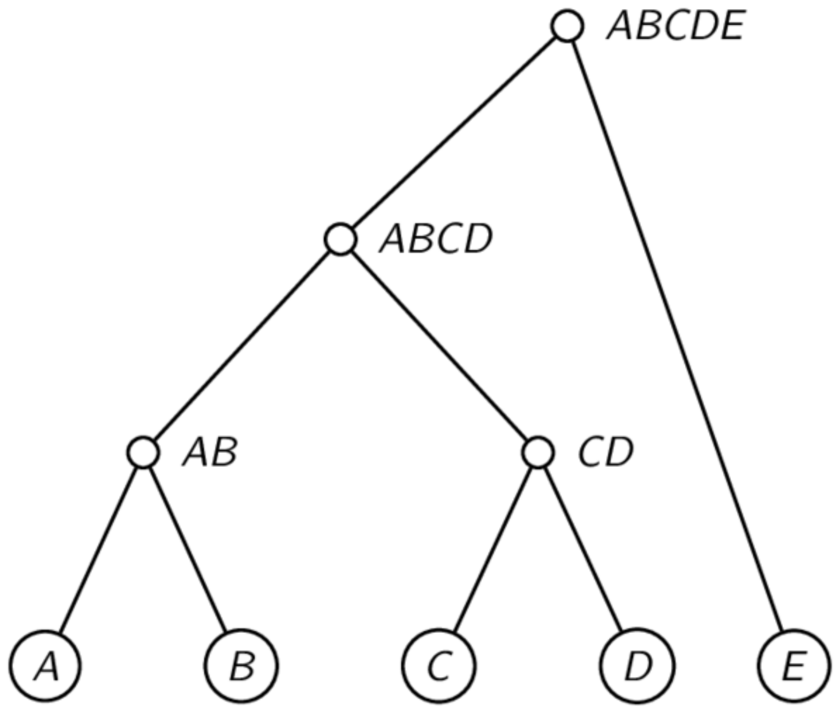
Hierarchical Clustering: Nested Partitions

Given a dataset $\boldsymbol{D} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}$, where $\boldsymbol{x}_i \in \mathbb{R}^d$, a clustering $\mathcal{C} = \{C_1, \ldots, C_k\}$ is a partition of $\boldsymbol{D}$.

A clustering $\mathcal{A} = \{A_1, \ldots, A_r\}$ is said to be nested in another clustering $\mathcal{B} = \{B_1, \ldots, B_s\}$ if and only if $r > s$, and for each cluster $A_i \in \mathcal{A}$, there exists a cluster $B_j \in \mathcal{B}$, such that $A_i \subseteq B_j$.

Hierarchical clustering yields a sequence of n nested partitions $\mathcal{C}_1, \ldots, \mathcal{C}_n$. The clustering $\mathcal{C}_{t-1}$ is nested in the clustering $\mathcal{C}_t$. The cluster dendrogram is a rooted binary tree that captures this nesting structure, with edges between cluster $C_i \in \mathcal{C}_{t-1}$ and cluster $C_j \in \mathcal{C}_t$ if C_i is nested in C_j, that is, if $C_i \subset C_j$.

Hierarchical Clustering Dendrogram



The dendrogram represents the following sequence of nested partitions:

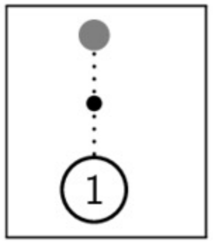
Clustering	**Clusters**
$\mathcal{C}_1$	$\{A\},\{B\},\{C\},\{D\},\{E\}$
$\mathcal{C}_2$	$\{AB\},\{C\},\{D\},\{E\}$
$\mathcal{C}_3$	$\{AB\},\{CD\},\{E\}$
$\mathcal{C}_4$	$\{ABCD\},\{E\}$
$\mathcal{C}_5$	$\{ABCDE\}$

with $\mathcal{C}_{t-1} \subset \mathcal{C}_t$ for $t = 2,\ldots,5$. We assume that A and B are merged before C and D.

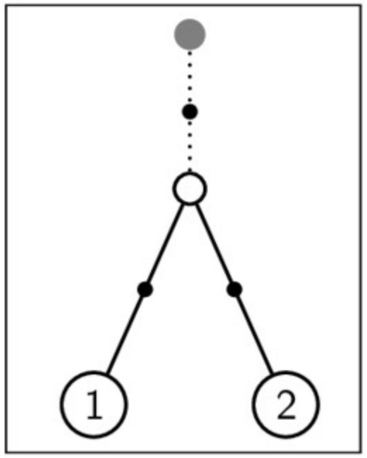
Number of Hierarchical Clusterings

The total number of different dendrograms with n leaves is given as:

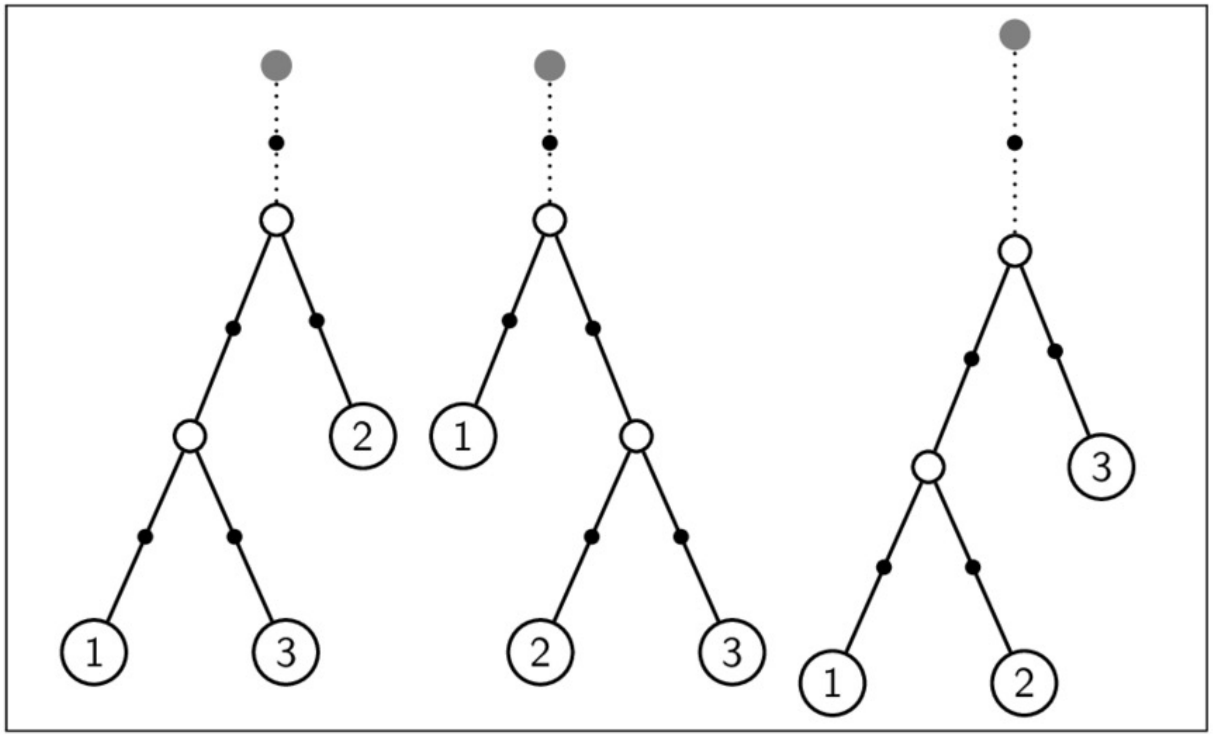
$$\prod_{m=1}^{n-1}(2m-1) = 1 \times 3 \times 5 \times 7 \times \cdots \times (2n-3) = (2n-3)!!$$



(a) $n = 1$



(b) $n = 2$



(c) $n = 3$

Agglomerative Hierarchical Clustering

In agglomerative hierarchical clustering, we begin with each of the n points in a separate cluster. We repeatedly merge the two closest clusters until all points are members of the same cluster.

Given a set of clusters $\mathcal{C} = \{C_1, C_2, .., C_m\}$, we find the *closest* pair of clusters C_i and C_j and merge them into a new cluster $C_{ij} = C_i \cup C_j$.

Next, we update the set of clusters by removing C_i and C_j and adding C_{ij}, as follows $\mathcal{C} = \big(\mathcal{C} \setminus \{C_i, C_j\}\big) \cup \{C_{ij}\}$.

We repeat the process until $\mathcal{C}$ contains only one cluster. If specified, we can stop the merging process when there are exactly k clusters remaining.

Agglomerative Hierarchical Clustering Algorithm

```
AgglomerativeClustering($\boldsymbol{D}, k$):
1 $\mathcal{C} \leftarrow \{C_i = \{\boldsymbol{x}_i\} \mid \boldsymbol{x}_i \in \boldsymbol{D}\}$ // Each point in separate cluster
2 $\Delta \leftarrow \{\delta(\boldsymbol{x}_i, \boldsymbol{x}_j) \colon \boldsymbol{x}_i, \boldsymbol{x}_j \in \boldsymbol{D}\}$ // Compute distance matrix
3 repeat
4     Find the closest pair of clusters $C_i, C_j \in \mathcal{C}$
5     $C_{ij} \leftarrow C_i \cup C_j$ // Merge the clusters
6     $\mathcal{C} \leftarrow \big(\mathcal{C} \setminus \{C_i, C_j\}\big) \cup \{C_{ij}\}$ // Update the clustering
7     Update distance matrix $\Delta$ to reflect new clustering
8 until $|\mathcal{C}| = k$
```

Distance between Clusters

Single, Complete and Average

A typical distance between two points is the Euclidean distance or L_2-*norm*

$$\|\mathbf{x}-\mathbf{y}\|_2 = \left(\sum_{i=1}^{d}(x_i - y_i)^2\right)^{1/2}$$

Single Link: The minimum distance between a point in C_i and a point in C_j

$$\boxed{\delta(C_i, C_j) = \min\{\|\mathbf{x}-\mathbf{y}\| \mid \mathbf{x} \in C_i, \mathbf{y} \in C_j\}}$$

Complete Link: The maximum distance between points in the two clusters:

$$\boxed{\delta(C_i, C_j) = \max\{\|\mathbf{x}-\mathbf{y}\| \mid \mathbf{x} \in C_i, \mathbf{y} \in C_j\}}$$

Group Average: The average pairwise distance between points in C_i and C_j:

$$\boxed{\delta(C_i, C_j) = \frac{\sum_{\mathbf{x}\in C_i}\sum_{\mathbf{y}\in C_j}\|\mathbf{x}-\mathbf{y}\|}{n_i \cdot n_j}}$$

Distance between Clusters: Mean and Ward's

Mean Distance: The distance between two clusters is defined as the distance between the means or centroids of the two clusters:

$$\boxed{\delta(C_i, C_j) = \left\| \boldsymbol{\mu}_i - \boldsymbol{\mu}_j \right\|}$$

Minimum Variance or Ward's Method: The distance between two clusters is defined as the increase in the sum of squared errors (SSE) when the two clusters are merged, where the SSE for a given cluster C_i is given as

$$\delta(C_i, C_j) = \Delta SSE_{ij} = SSE_{ij} - SSE_i - SSE_j$$

where $SSE_i = \sum_{\mathbf{x} \in C_i} \|\mathbf{x} - \boldsymbol{\mu}_i\|^2$. After simplification, we get:

$$\boxed{\delta(C_i, C_j) = \left(\frac{n_i n_j}{n_i + n_j} \right) \left\| \boldsymbol{\mu}_i - \boldsymbol{\mu}_j \right\|^2}$$

Ward's measure is therefore a weighted version of the mean distance measure.

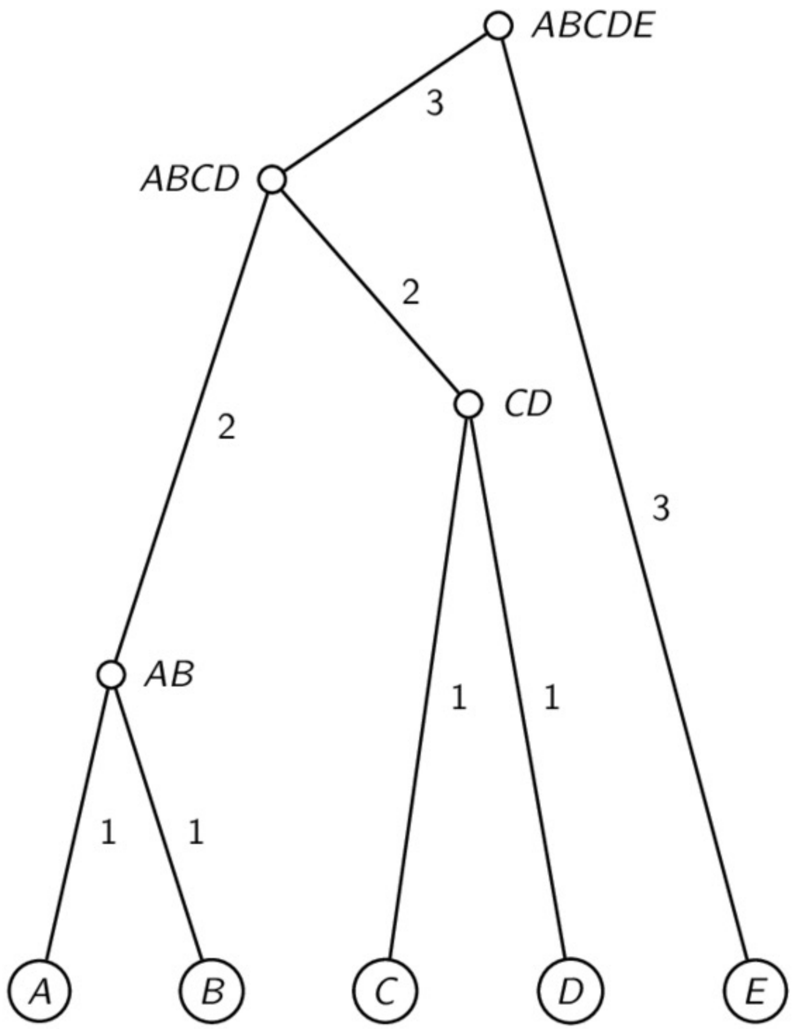
Single Link Agglomerative Clustering

δ	E
$ABCD$	③

δ	CD	E
AB	②	3
CD		3

δ	C	D	E
AB	3	2	3
C		①	3
D			5

δ	B	C	D	E
A	①	3	2	4
B		3	2	3
C			1	3
D				5



Lance–Williams Formula

Whenever two clusters C_i and C_j are merged into C_{ij}, we need to update the distance matrix by recomputing the distances from the newly created cluster C_{ij} to all other clusters C_r ($r \neq i$ and $r \neq j$).

The Lance–Williams formula provides a general equation to recompute the distances for all of the cluster proximity measures

$$\begin{aligned}\delta(C_{ij}, C_r) = \alpha_i \cdot \delta(C_i, C_r) + \alpha_j \cdot \delta(C_j, C_r) + \\ \beta \cdot \delta(C_i, C_j) + \gamma \cdot \left|\delta(C_i, C_r) - \delta(C_j, C_r)\right|\end{aligned}$$

The coefficients $\alpha_i, \alpha_j, \beta$, and γ differ from one measure to another.

Lance–Williams Formulas for Cluster Proximity

Measure	α_i	α_j	β	γ
Single link	$\frac{1}{2}$	$\frac{1}{2}$	0	$-\frac{1}{2}$
Complete link	$\frac{1}{2}$	$\frac{1}{2}$	0	$\frac{1}{2}$
Group average	$\frac{n_i}{n_i+n_j}$	$\frac{n_j}{n_i+n_j}$	0	0
Mean distance	$\frac{n_i}{n_i+n_j}$	$\frac{n_j}{n_i+n_j}$	$\frac{-n_i \cdot n_j}{(n_i+n_j)^2}$	0
Ward's measure	$\frac{n_i+n_r}{n_i+n_j+n_r}$	$\frac{n_j+n_r}{n_i+n_j+n_r}$	$\frac{-n_r}{n_i+n_j+n_r}$	0

Lance–Williams Formulas for Cluster Proximity

Single link: Arithmetical trick to find the minimum.

$$\delta(C_{ij}, C_r) = \frac{\delta(C_i, C_r)}{2} + \frac{\delta(C_j, C_r)}{2} - \frac{\left|\delta(C_i, C_r) - \delta(C_j, C_r)\right|}{2}$$

Complete link: Arithmetical trick to find the maximum.

$$\delta(C_{ij}, C_r) = \frac{\delta(C_i, C_r)}{2} + \frac{\delta(C_j, C_r)}{2} + \frac{\left|\delta(C_i, C_r) - \delta(C_j, C_r)\right|}{2}$$

Group average: Weight the distance by the cluster size.

$$\delta(C_{ij}, C_r) = \frac{n_i}{n_i + n_j} \cdot \delta(C_i, C_r) + \frac{n_j}{n_i + n_j} \cdot \delta(C_j, C_r)$$

Lance–Williams Formulas for Cluster Proximity

Mean distance: The new centroid is in the line defined by μ_i and μ_j, and its distance to μ_r has to be adjusted by $\frac{n_i \cdot n_j}{(n_i+n_j)^2}$.

$$\delta(C_{ij}, C_r) = \frac{n_i}{n_i + n_j} \cdot \delta(C_i, C_r) + \frac{n_j}{n_i + n_j} \cdot \delta(C_j, C_r) + \frac{-n_i \cdot n_j}{(n_i + n_j)^2} \cdot \delta(C_i, C_j)$$

Ward's measure: The ΔSSE of the new cluster is a weigthed sum of the $\Delta SSEs$ of the original clusters, adjusted by the fact that n_r was considered twice.

$$\delta(C_{ij}, C_r) = \frac{n_i + n_r}{n_i + n_j + n_r} \cdot \delta(C_i, C_r) + \frac{n_j + n_r}{n_i + n_j + n_r} \cdot \delta(C_j, C_r) + \frac{-n_r}{n_i + n_j + n_r} \cdot \delta(C_i, C_j)$$

Lance-Williams: Single Link

- Single link distance $\delta(C_i, C_j) = \min\{\delta(x,y): x \in C_i, y \in C_j\}$
- Check Lance-Williams Formula for Single Link

$$\delta(C_{ij}, C_r) = ½\, \delta(C_i, C_r) + ½\, \delta(C_j, C_r) - ½\, |\, \delta(C_i, C_r) - \delta(C_j, C_r)\, |$$

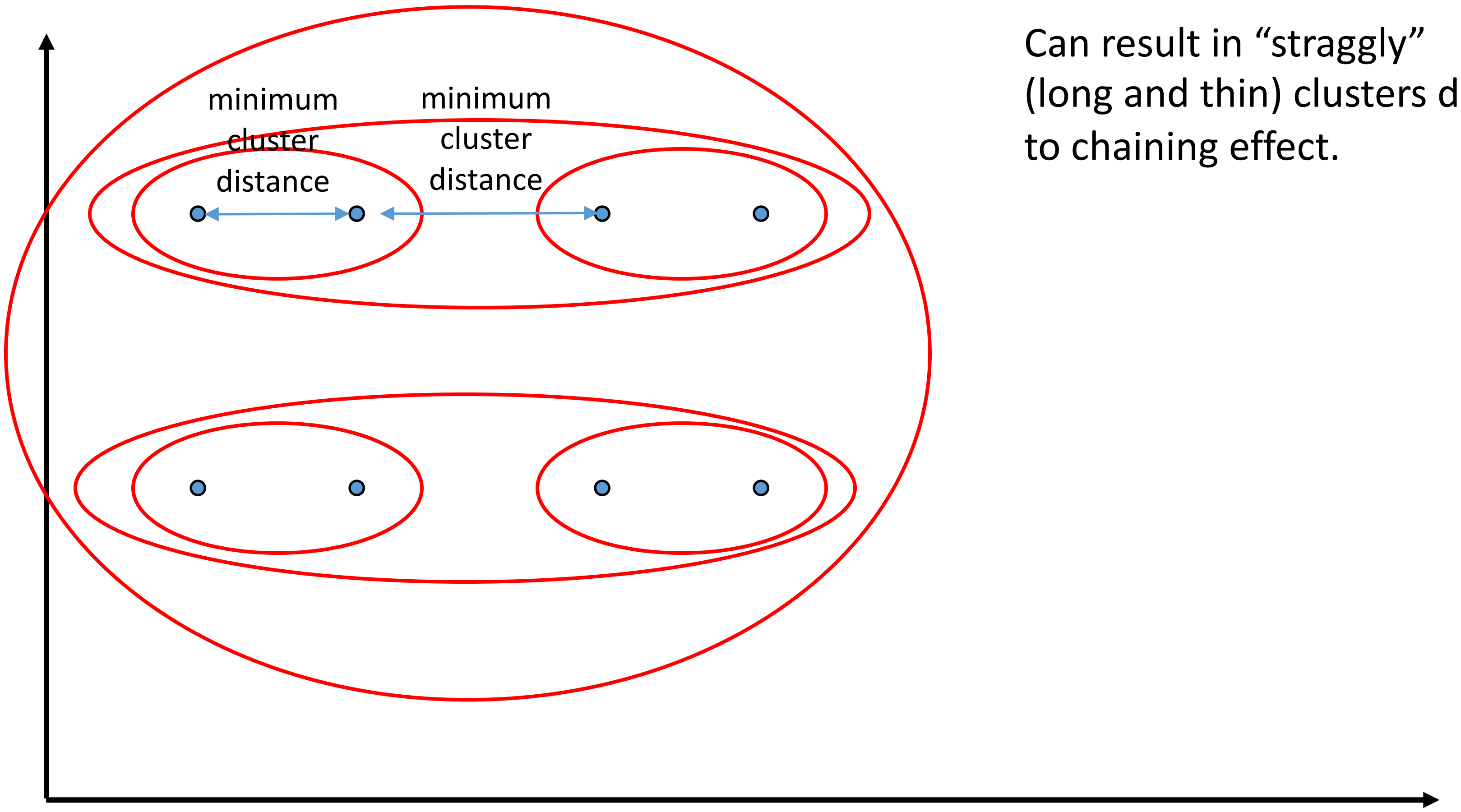
- $\delta(C_{ij}, C_r)$

$$\begin{aligned} &= \min\{\delta(x,y): x \in C_{ij}, y \in C_r\} \\ &= \min\{\delta(x,y): x \in C_i \cup C_j, y \in C_r\} \\ &= \min\{\min\{\delta(x,y): x\in C_i, y\in C_r\}, \min\{\delta(x,y): x \in C_j, y \in C_r\}\} \\ &= \min\{\delta(C_i, C_r), \delta(C_j, C_r)\} \end{aligned}$$

Same as formula above because

$$\min(a,b) = a/2 + b/2 - |\, a - b\, |/2 \quad \text{(Check cases } a > b \text{ and } a < b\text{)}$$

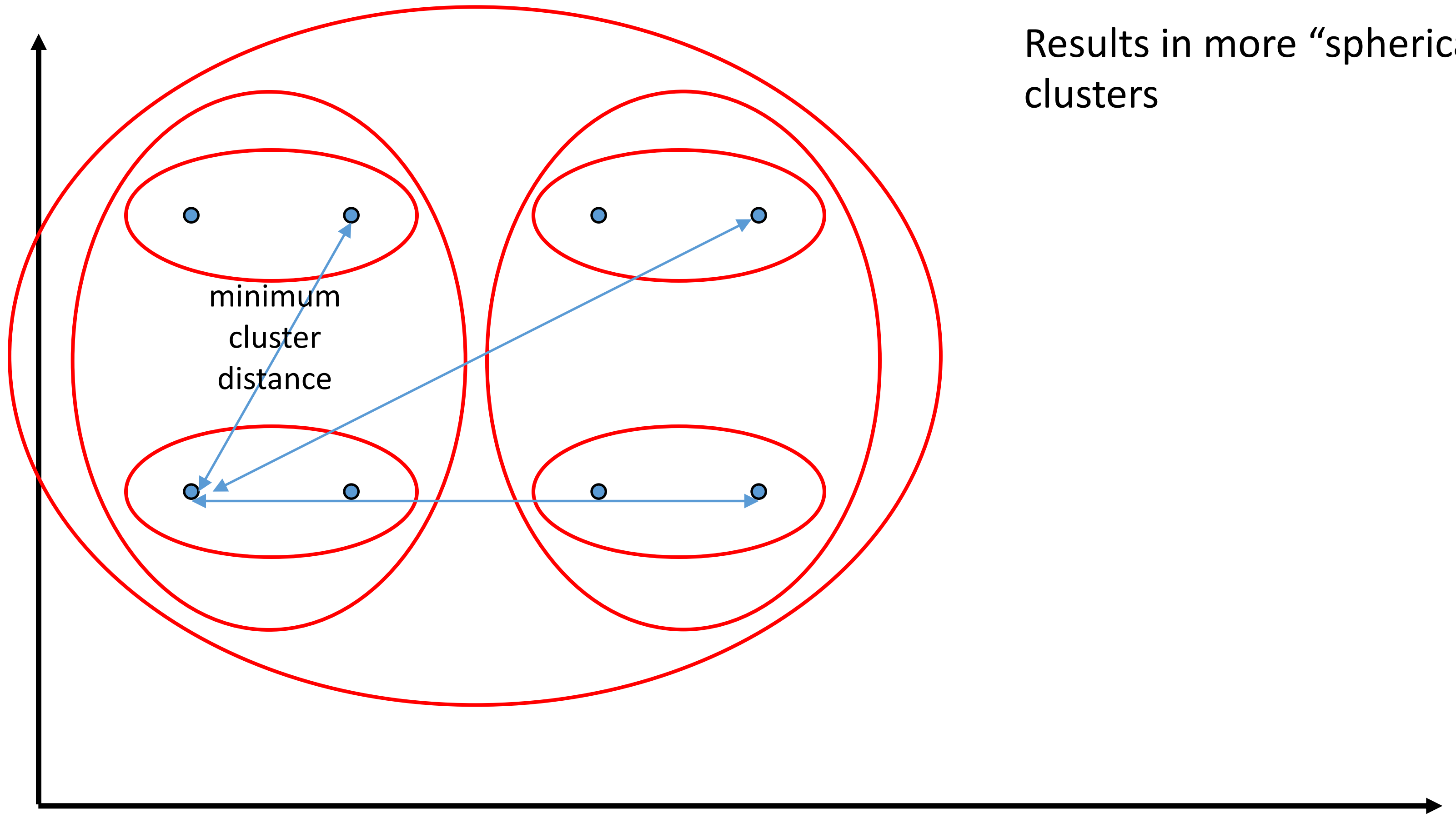
Single Link Example



Can result in "straggly" (long and thin) clusters due to chaining effect.

$$\delta(C_i, C_j) = \min\{\delta(\mathbf{x}, \mathbf{y}) \mid \mathbf{x} \in C_i, \mathbf{y} \in C_j\}$$

Complete Link Example



Results in more “spherical” clusters

$$\delta(C_i, C_j) = \max\{\delta(\mathbf{x}, \mathbf{y}) \mid \mathbf{x} \in C_i, \mathbf{y} \in C_j\}$$

Computational Complexity for Hierarchical Clustering

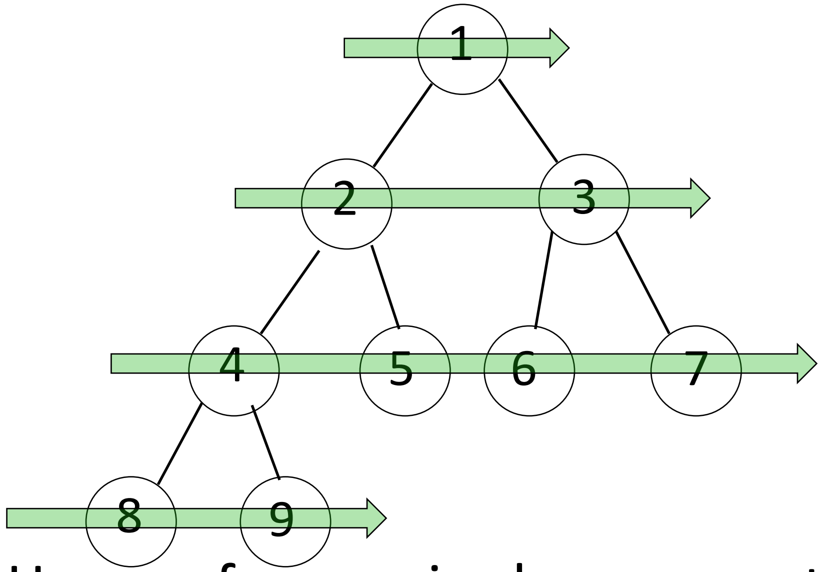
- N items
- Space complexity:
  - N items require O(N^2) distance matrix
- Time complexity:
  - Naïve approach:
    - Each of N steps, find minimum of O(N^2)distances: O(N^3)
    - Very slow for large N
- Is there a better approach to find minimum distance
  - Can we store the distance matrix in a way that makes it easier to find the minimum at each step

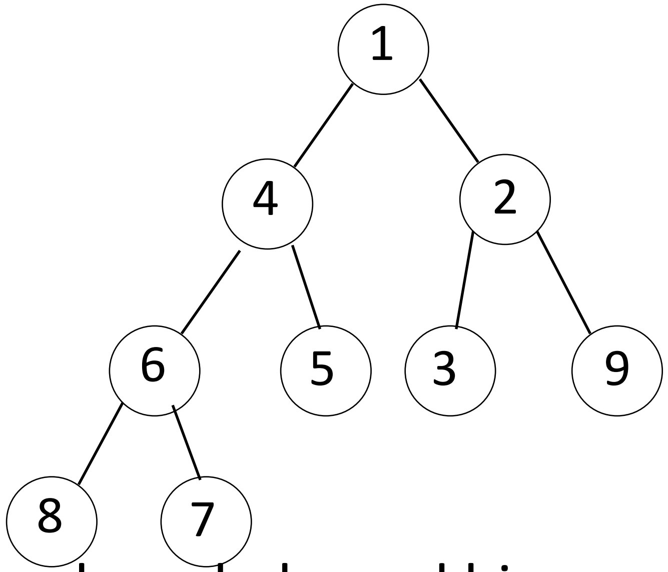
Heap Data Structure

- Aim:
  - Store a set of numbers in a way that makes it easy to keep track of minimum
  - Property preserved under insertions and deletions
- Benefit:
  - Don’t need to repeat work of finding minimum after insertions and deletions

Heaps

- Min Heap
  - Each node has at most two children
  - Parents have smaller entry than children
  - Each depth of is filled in order (top to bottom, left to right)
- Two possible heaps storing the numbers {1,..9}



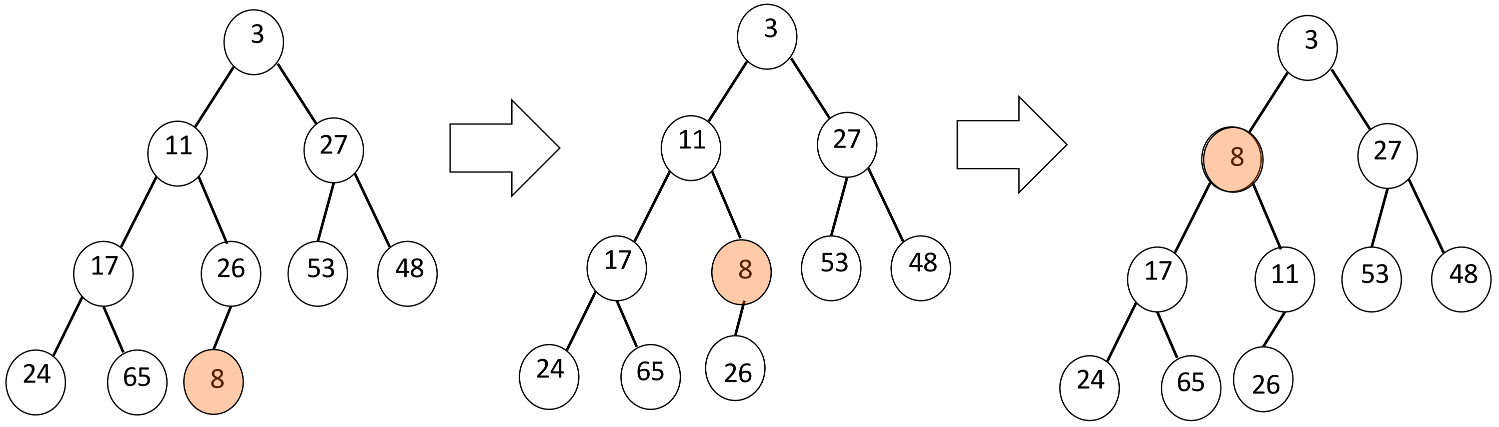


Heaps of same size have same topology; balanced binary

Insertion into Heap

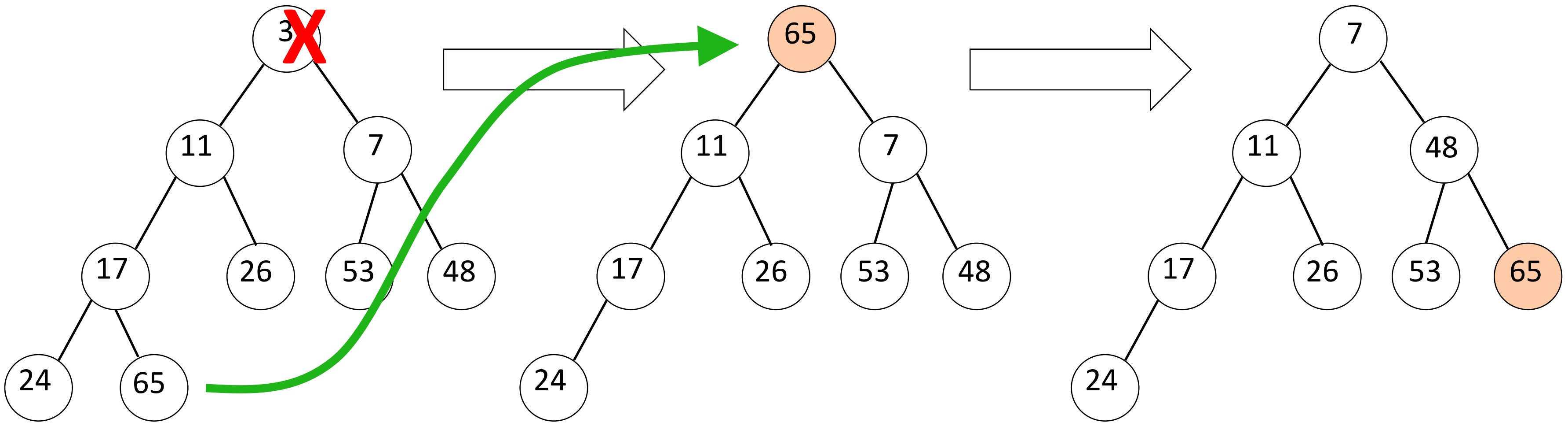
- Insert new item at next free position
- Bubble up by swapping with until a heap is obtained
- Have to do at most h swaps, where h = O(log n) is tree depth

- Insert(8)



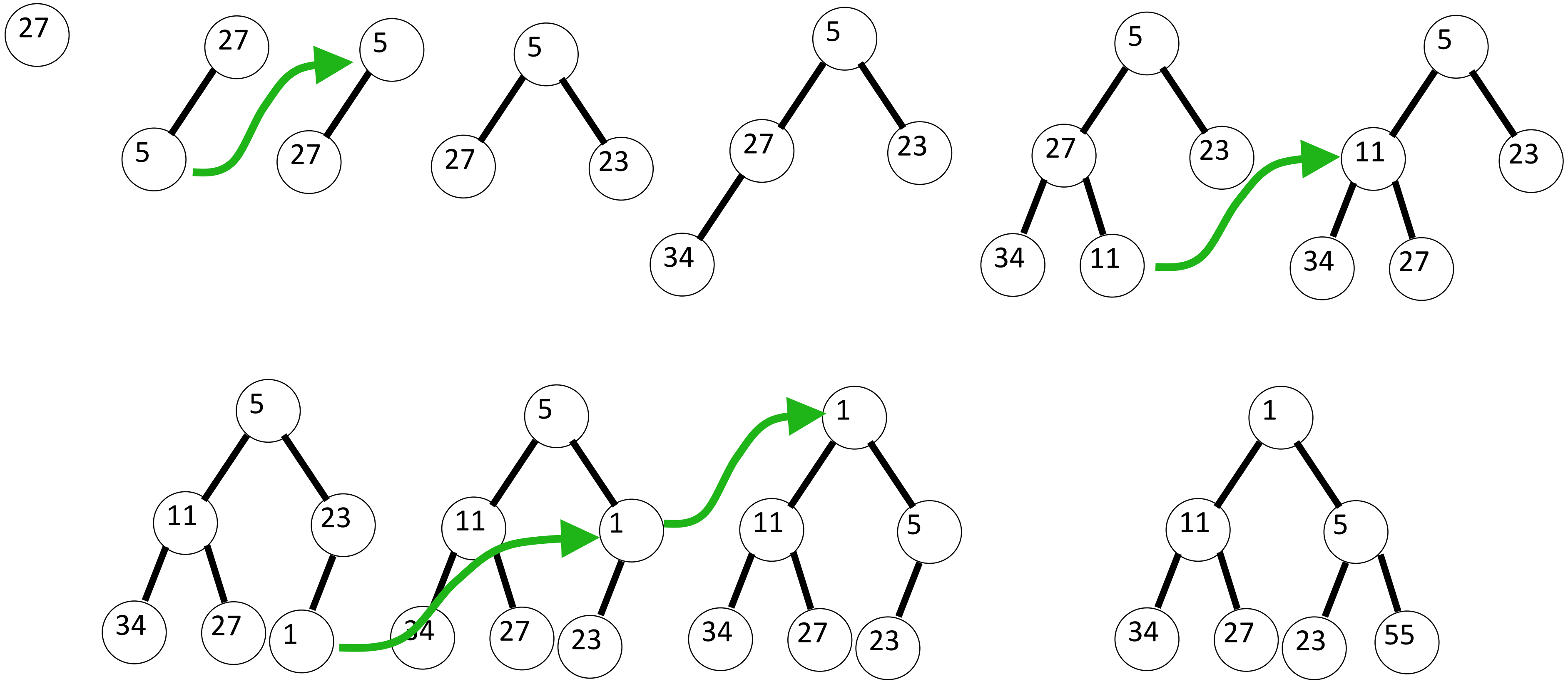
Deletion from heap

- Only the root node is removable
    - Smallest item when implementing a (low)-priority queue
- Remove root node
- Move node in last position to root
- Bubble down by swapping with smaller child until heap formed
    - Takes at most h swaps, where h = O(log n) is tree depth
- Delete(3)



Example: Initializing a Heap

- Insert in order 27, 5, 23, 34, 11, 1, 55



Complexity for Hierarchical Clustering

- Assume n initial points
- Initialization
  - $O(n^2)$ pairs, each costing $O(\log n)$ heap insertion
- Each of n iterations
  - Find minimum link: $O(1)$
  - Suppose amalgamate clusters C_i and C_j into C_{ij}
    - $O(n)$ deletions of distance $\delta(C_i, C_k)$ and $\delta(C_j, C_k)$
      - Delete from heap: set distance to 0 and bubble up and out
    - $O(n)$ insertions of distance $\delta(C_{ij}, C_k)$
  - $O(n)$ deletions & insertions: each cost $O(\log n)$
- Total cost $O(n^2 \log n)$
  - Can do in $O(n^2)$ by other means for single-link clustering