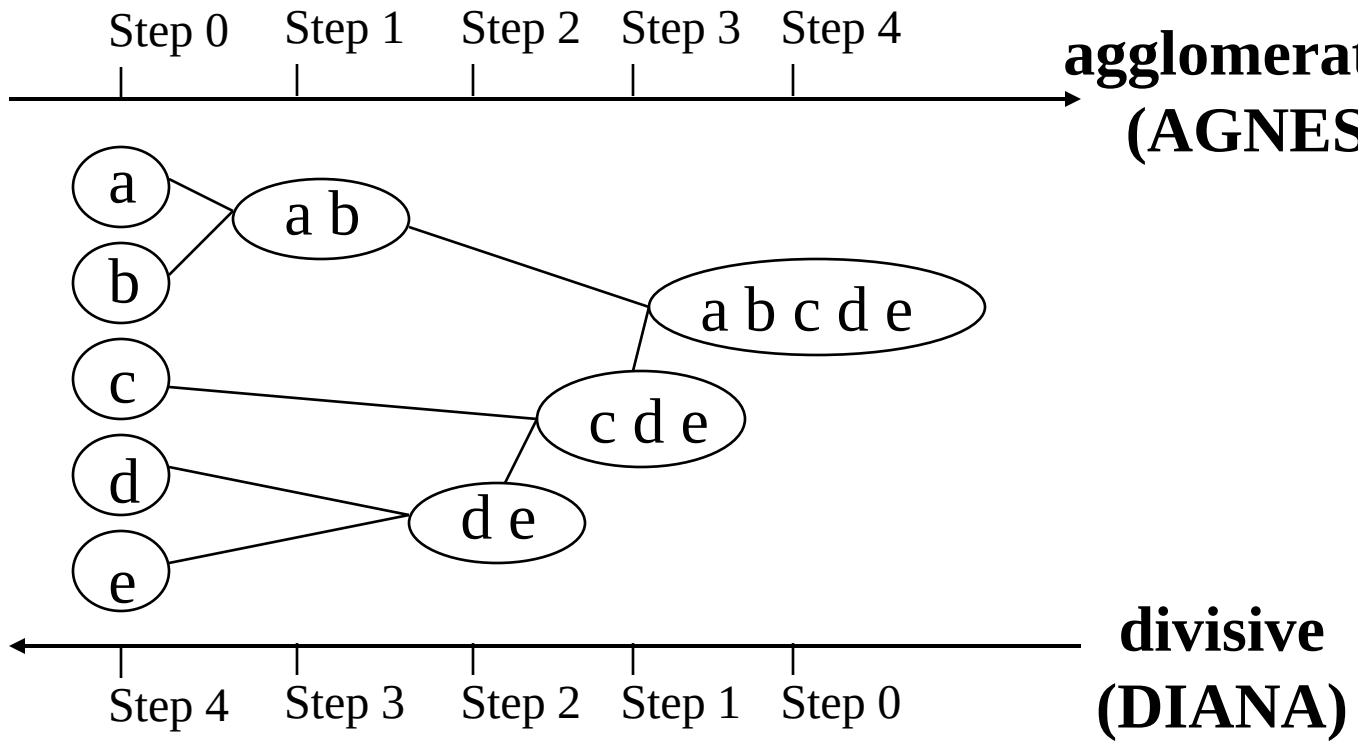
Cluster Analysis

Cluster Analysis

- What is Cluster Analysis?
- Types of Data in Cluster Analysis
- A Categorization of Major Clustering Methods
- Partitioning Methods
- Hierarchical Methods
- Density-Based Methods
- Grid-Based Methods
- Model-Based Clustering Methods
- Outlier Analysis
- Summary

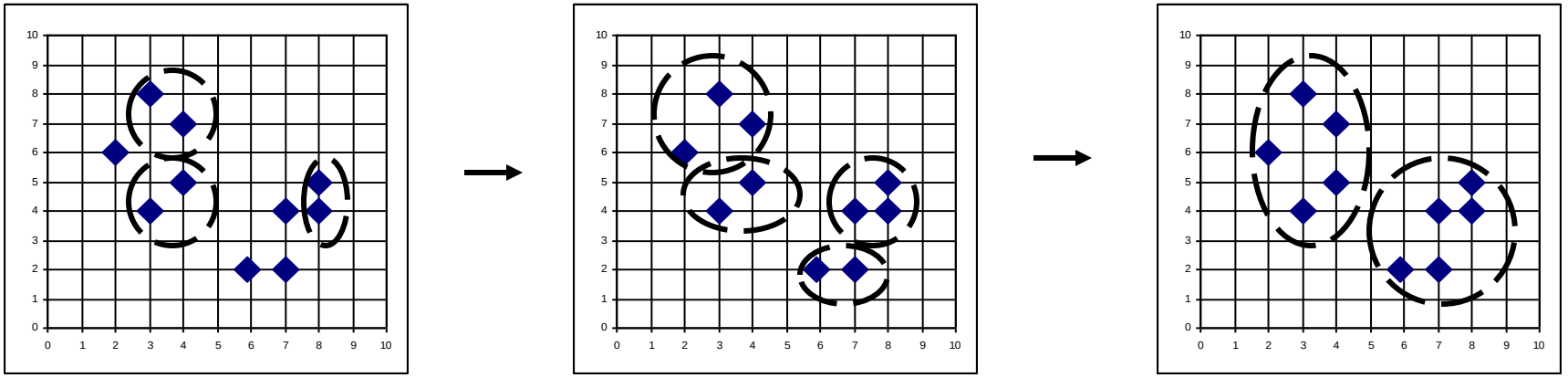
Hierarchical Clustering

- Use distance matrix as clustering criteria. This method does not require the number of clusters ***k*** as an input, but needs a termination condition



AGNES (Agglomerative Nesting)

- Implemented in statistical analysis packages, e.g., Splus
- Use the Single-Link method and the dissimilarity matrix.
- Merge objects that have the least dissimilarity
- Go on in a non-descending fashion
- Eventually all objects belong to the same cluster



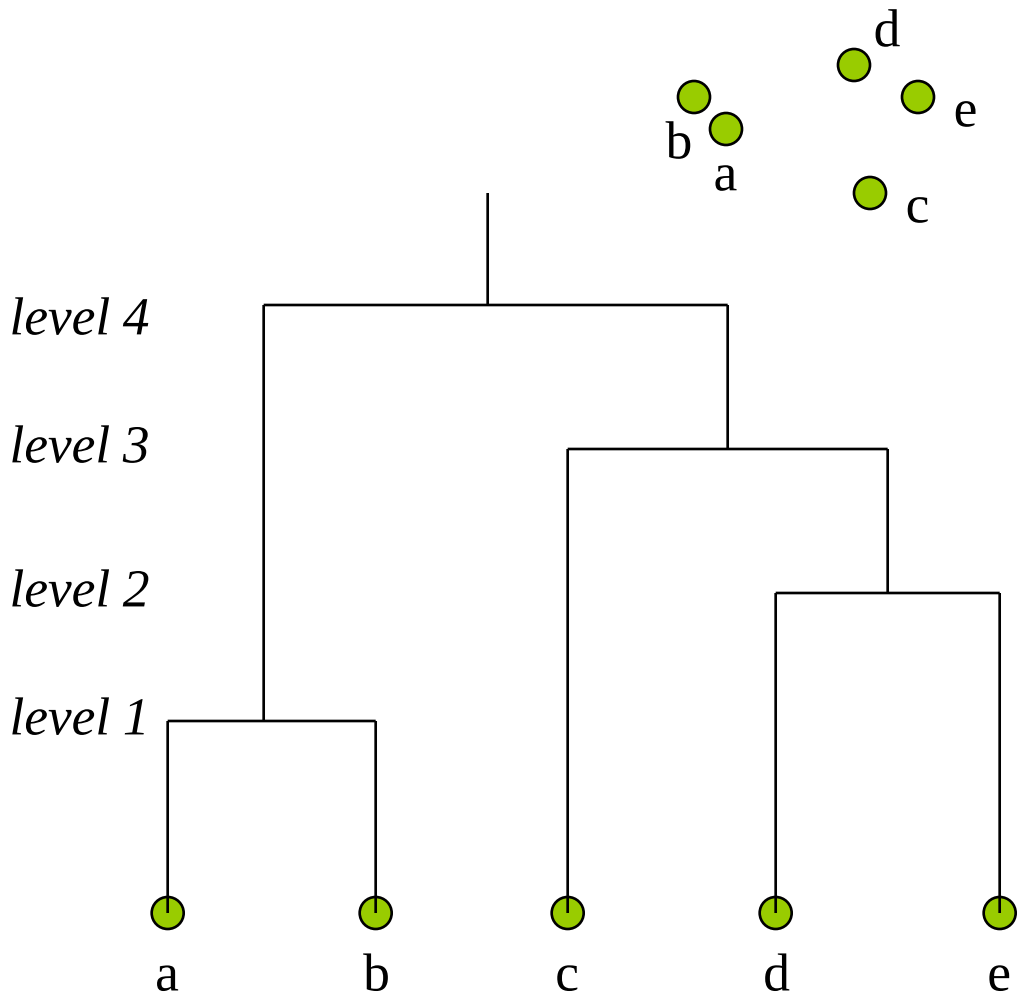
- Single-Link: each time merge the clusters (C_1,C_2) which are connected by the *shortest single link* of objects, i.e., $\min_{p \in C1, q \in C2} dist(p,q)$

A *Dendrogram* Shows How the Clusters are Merged Hierarchically

Decompose data objects into a several levels of nested partitioning (tree of clusters), called a dendrogram.

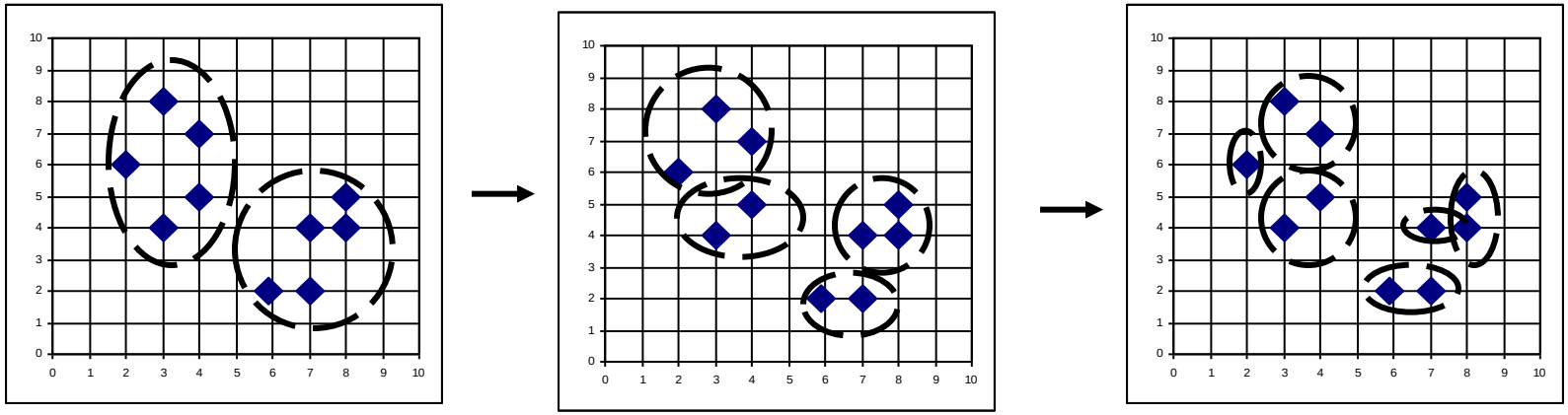
A clustering of the data objects is obtained by cutting the dendrogram at the desired level, then each connected component forms a cluster.

E.g., level 1 gives 4 clusters: {a,b},{c},{d},{e},
level 2 gives 3 clusters: {a,b},{c},{d,e}
level 3 gives 2 clusters: {a,b},{c,d,e}, etc.



DIANA (Divisive Analysis)

- Implemented in statistical analysis packages, e.g., Splus
- Inverse order of AGNES
- Eventually each node forms a cluster on its own



More on Hierarchical Clustering Methods

- Major weakness of agglomerative clustering methods
  - do not scale well: time complexity of at least $O(n^2)$, where n is the number of total objects
  - can never undo what was done previously
- Integration of hierarchical with distance-based clustering
  - BIRCH (1996): uses CF-tree and incrementally adjusts the quality of sub-clusters
  - CURE (1998): selects well-scattered points from the cluster and then shrinks them towards the center of the cluster by a specified fraction
  - CHAMELEON (1999): hierarchical clustering using dynamic modeling

BIRCH (1996)

- Birch: Balanced Iterative Reducing and Clustering using Hierarchies, by Zhang, Ramakrishnan, Livny (SIGMOD'96)
- Incrementally construct a CF (Clustering Feature) tree, a hierarchical data structure for multiphase clustering
  - Phase 1: scan DB to build an initial in-memory CF tree (a multi-level compression of the data that tries to preserve the inherent clustering structure of the data)
  - Phase 2: use an arbitrary clustering algorithm to cluster the leaf nodes of the CF-tree
- *Scales linearly*: finds a good clustering with a single scan and improves the quality with a few additional scans
- *Weakness:* handles only numeric data, and sensitive to the order of the data records, no good if non-spherical clusters.

Clustering Feature Vector

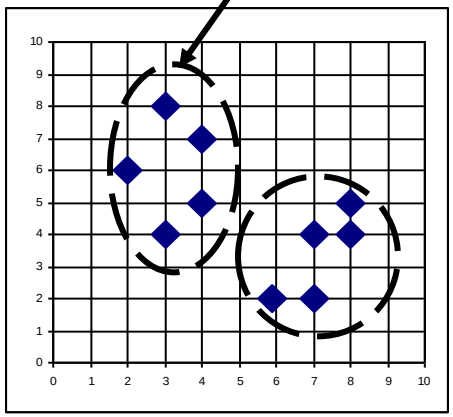
Clustering Feature: ***CF = (N, LS, SS)***

N: **Number of data points**

LS: $\sum_{i=1}^{N} \overrightarrow{X_i}$

SS: $\sum_{i=1}^{N} (\overrightarrow{X_i})^2$

CF = (5, (16,30),244)



(3,4)
(2,6)
(4,5)
(4,7)
(3,8)

Some Characteristics of CFVs

- Two CFVs can be aggregated.
  - Given CF1=(N1, LS1, SS1), CF2 = (N2, LS2, SS2),
  - If combined into one cluster, CF=(N1+N2, LS1+LS2, SS1+SS2).
- The centroid and radius can both be computed from CF.
  - centroid is the center of the cluster
  - radius is the average distance between an object and the centroid.

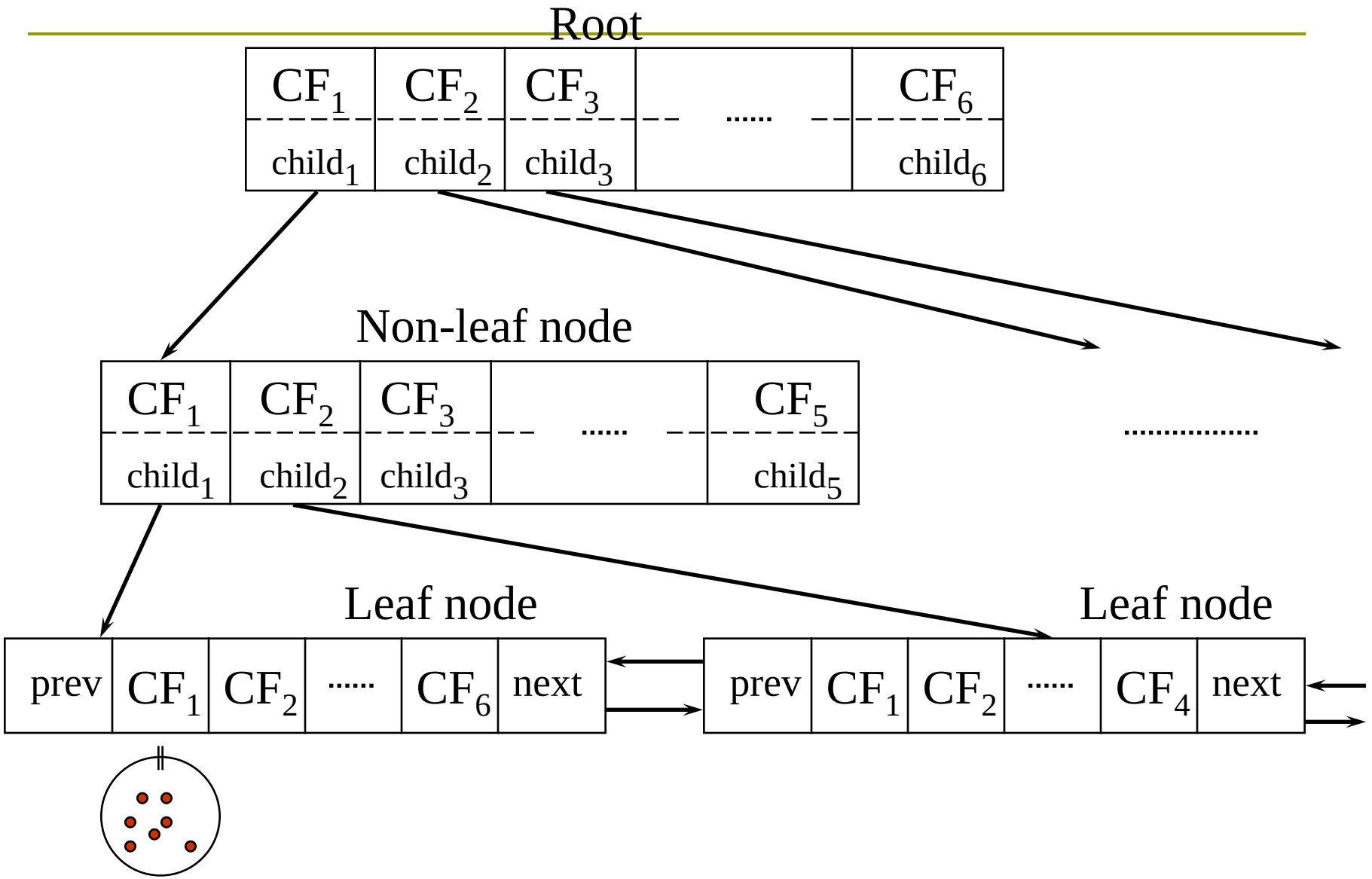
$$\vec{x}_0 = \frac{\sum_{i=1}^{N} \vec{x}_i}{N} \qquad R = \sqrt{\frac{\sum_{i=1}^{N} (\vec{x}_i - \vec{x}_0)^2}{N}}$$

Other statistical features as well...

CF-Tree in BIRCH

- A CF tree is a height-balanced tree that stores the clustering features for a hierarchical clustering
  - A nonleaf node in a tree has descendants or "children"
  - The nonleaf nodes store sums of the CFs of their children
- A CF tree has two parameters
  - Branching factor: specify the maximum number of children.
  - threshold T: max radius of sub-clusters stored at the leaf nodes

CF Tree (a multiway tree, like the B-tree)



CF-Tree Construction

- Scan through the database once.
- For each object, insert into the CF-tree as follows:
  - At each level, choose the sub-tree whose centroid is closest.
  - In a leaf page, choose a cluster that can absort it (new radius < T). If no cluster can absorb it, create a new cluster.
  - Update upper levels.