Data Mining:
Concepts and Techniques

— Chapter 7 —

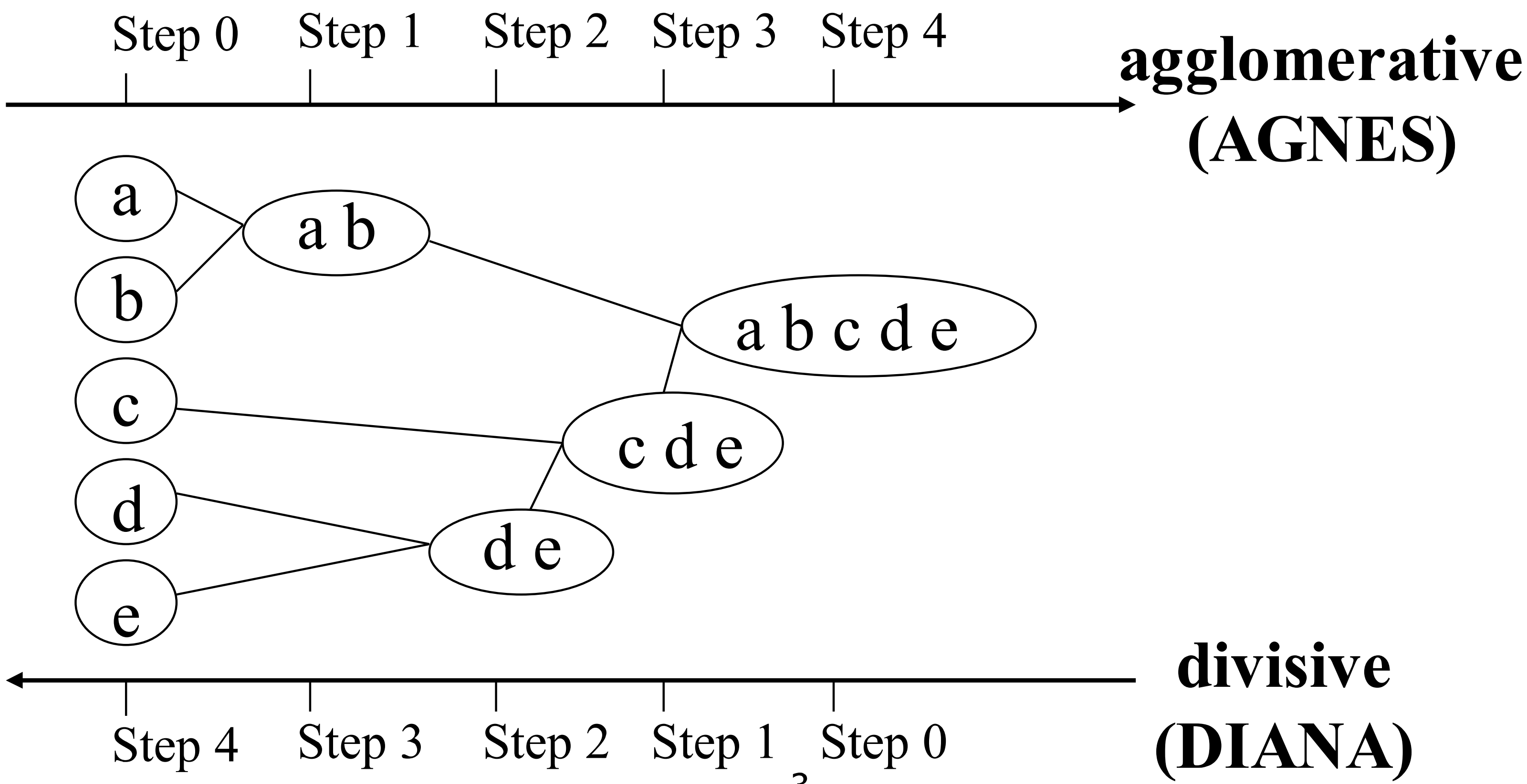
Jiawei Han
Department of Computer Science
University of Illinois at Urbana-Champaign
www.cs.uiuc.edu/~hanj

Cluster Analysis

1. What is Cluster Analysis?
2. Types of Data in Cluster Analysis
3. A Categorization of Major Clustering Methods
4. Partitioning Methods
5. Hierarchical Methods
6. Density-Based Methods

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*



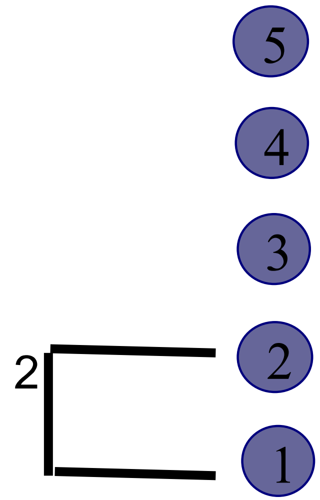
Complete-link Clustering Example

$$
\begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array}
\begin{array}{c} \begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \end{array} \\ \left[\begin{array}{ccccc} 0 & & & & \\ 2 & 0 & & & \\ 6 & 3 & 0 & & \\ 10 & 9 & 7 & 0 & \\ 9 & 8 & 5 & 4 & 0 \end{array}\right] \end{array}
\Rightarrow
\begin{array}{c} \\ (1,2) \\ 3 \\ 4 \\ 5 \end{array}
\begin{array}{c} \begin{array}{cccc} (1,2) & 3 & 4 & 5 \end{array} \\ \left[\begin{array}{cccc} 0 & & & \\ 6 & 0 & & \\ 10 & 7 & 0 & \\ 9 & 5 & 4 & 0 \end{array}\right] \end{array}
$$

$$d_{(1,2),3} = \max\{d_{1,3}, d_{2,3}\} = \max\{6,3\} = 6$$

$$d_{(1,2),4} = \max\{d_{1,4}, d_{2,4}\} = \max\{10,9\} = 10$$

$$d_{(1,2),5} = \max\{d_{1,5}, d_{2,5}\} = \max\{9,8\} = 9$$

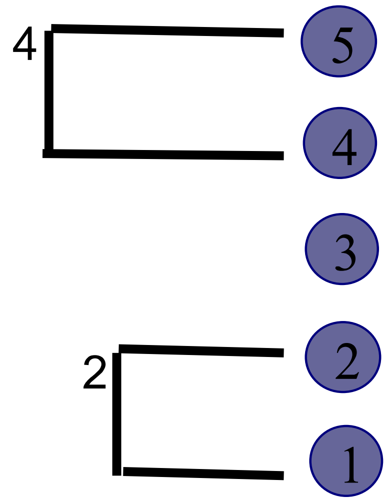


Complete-link Clustering Example

$$
\begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array}
\begin{array}{c} \begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \end{array} \\ \left[\begin{array}{ccccc} 0 & & & & \\ 2 & 0 & & & \\ 6 & 3 & 0 & & \\ 10 & 9 & 7 & 0 & \\ 9 & 8 & 5 & 4 & 0 \end{array}\right] \end{array}
\Rightarrow
\begin{array}{c} \\ (1,2) \\ 3 \\ 4 \\ 5 \end{array}
\begin{array}{c} \begin{array}{cccc} (1,2) & 3 & 4 & 5 \end{array} \\ \left[\begin{array}{cccc} 0 & & & \\ 6 & 0 & & \\ 10 & 7 & 0 & \\ 9 & 5 & 4 & 0 \end{array}\right] \end{array}
\Rightarrow
\begin{array}{c} \\ (1,2) \\ 3 \\ (4,5) \end{array}
\begin{array}{c} \begin{array}{ccc} (1,2) & 3 & (4,5) \end{array} \\ \left[\begin{array}{ccc} 0 & & \\ 6 & 0 & \\ 10 & 7 & 0 \end{array}\right] \end{array}
$$

$$d_{(1,2),(4,5)} = \max\{d_{(1,2),4}, d_{(1,2),5}\} = \max\{10,9\} = 10$$

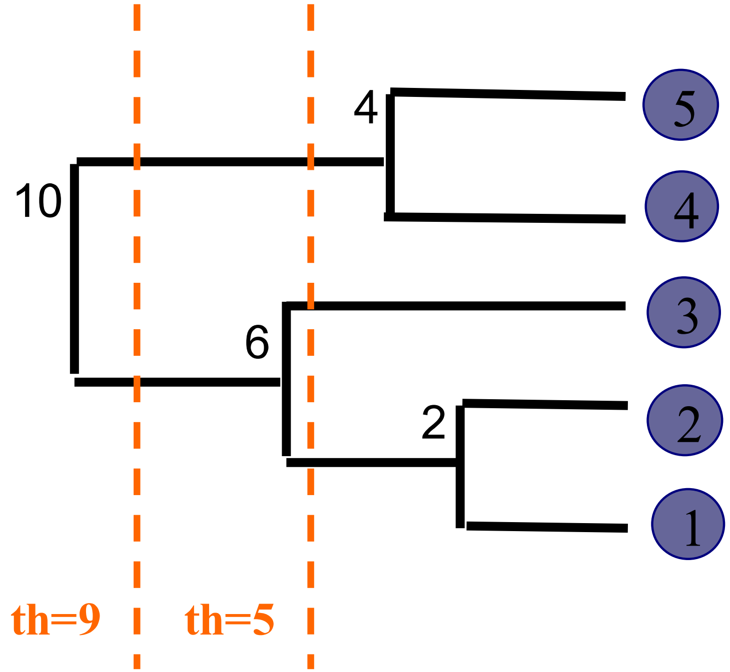
$$d_{3,(4,5)} = \max\{d_{3,4}, d_{3,5}\} = \max\{7,5\} = 7$$



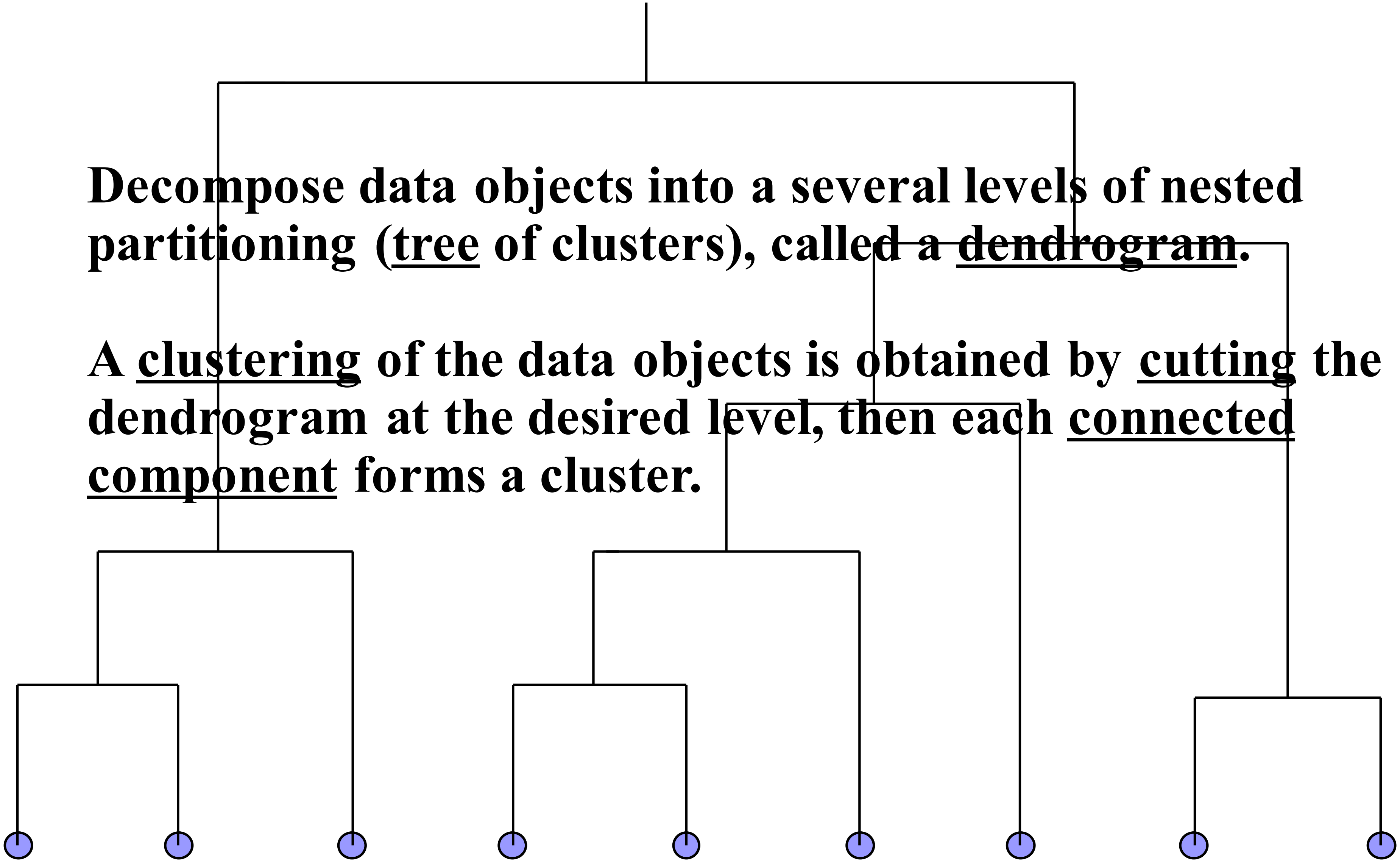
Complete-link Clustering Example

$$
\begin{array}{c}
 \\ 1 \\ 2 \\ 3 \\ 4 \\ 5
\end{array}
\begin{array}{c}
\begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \end{array} \\
\begin{bmatrix}
0 & & & & \\
2 & 0 & & & \\
6 & 3 & 0 & & \\
10 & 9 & 7 & 0 & \\
9 & 8 & 5 & 4 & 0
\end{bmatrix}
\end{array}
\Rightarrow
\begin{array}{c|cccc}
 & (1,2) & 3 & 4 & 5 \\
\hline
(1,2) & 0 & & & \\
3 & 6 & 0 & & \\
4 & 10 & 7 & 0 & \\
5 & 9 & 5 & 4 & 0
\end{array}
\Rightarrow
\begin{array}{c|ccc}
 & (1,2) & 3 & (4,5) \\
\hline
(1,2) & 0 & & \\
3 & 6 & 0 & \\
(4,5) & 10 & 7 & 0
\end{array}
$$

$$d_{(1,2,3),(4,5)} = \max\{ d_{(1,2),(4,5)}, d_{3,(4,5)} \} = 10$$



Dendrogram: Shows How the Clusters are Merged



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

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

AGNES (Agglomerative Nesting)

- Introduced in Kaufmann and Rousseeuw (1990)
- Implemented in statistical analysis packages, e.g., Splus
- Use the Single-Link method and the dissimilarity matrix.
- Merge nodes that have the least dissimilarity
- Go on in a non-descending fashion
- Eventually all nodes belong to the same cluster

DIANA (Divisive Analysis)

- Introduced in Kaufmann and Rousseeuw (1990)
- Implemented in statistical analysis packages, e.g., Splus
- Inverse order of AGNES
- Eventually each node forms a cluster on its own

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
  - ROCK (1999): clustering categorical data by neighbor and link analysis
  - CHAMELEON (1999): hierarchical clustering using dynamic modeling

BIRCH

- Birch: Balanced Iterative Reducing and Clustering using Hierarchies (Zhang, Ramakrishnan & Livny, 1996)
- 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 record, not always natural clusters.

Clustering Feature Vector in BIRCH

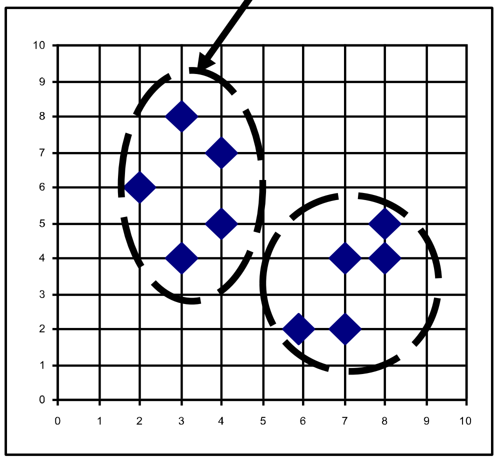
Clustering Feature: $CF = (N, LS, SS)$

N: **Number of data points**

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

SS: $\sum^{N}_{i=1} = \overrightarrow{X_i^2}$

CF = (5, (16,30),(54,190))

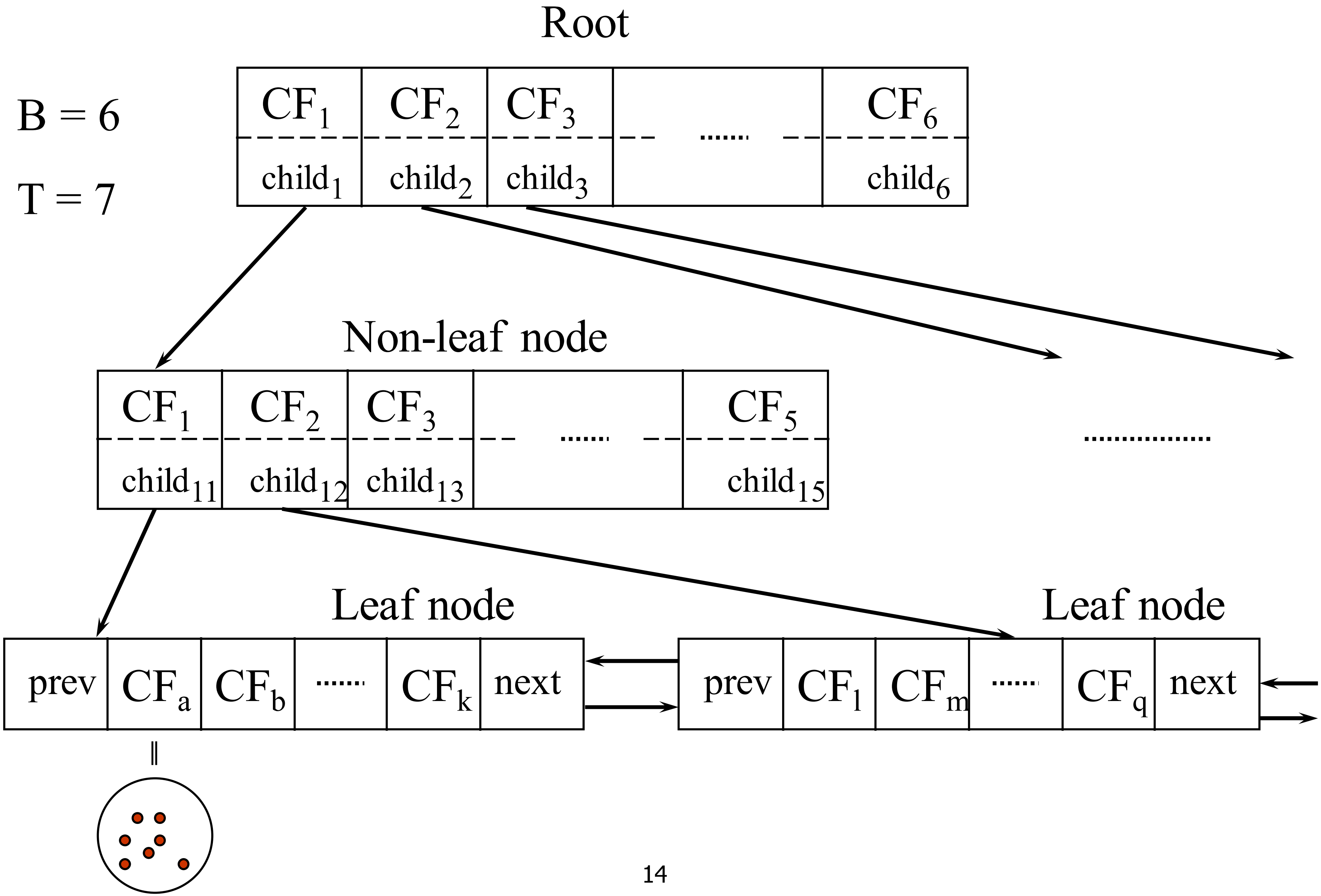


(3,4)
(2,6)
(4,5)
(4,7)
(3,8)

CF-Tree in BIRCH

- Clustering feature:
  - summary of the statistics for a given subcluster: the 0-th, 1st and 2nd moments of the subcluster from the statistical point of view.
  - registers crucial measurements for computing cluster and utilizes storage efficiently
- A CF tree is a height-balanced tree that stores the clustering features for a hierarchical clustering
  - A nonleaf node in a tree has children and stores the sums of the CFs of their children
  - A nonleaf node represents a cluster made of the subclusters represented by its children
  - A leaf node represents a cluster made of the subclusters represented by its entries
- A CF tree has two parameters
  - Branching factor: specify the maximum number of children.
  - threshold: max diameter of sub-clusters stored at the leaf nodes

The CF Tree Structure



Explanation on whiteboard

Example on whiteboard

ROCK: Clustering Categorical Data

- ROCK: RObust Clustering using linKs
  - S. Guha, R. Rastogi & K. Shim, 1999
- Major ideas
  - Use links to measure similarity/proximity

    maximize the sum of the number of links between points within a cluster, minimize the sum of the number of links for points in different clusters
  - Computational complexity:

$$O(n^2 + nm_m m_a + n^2 \log n)$$

Similarity Measure in ROCK

- Traditional measures for *categorical data* may not work well, e.g., Jaccard coefficient

Example: Two groups (clusters) of transactions

- C_1. <a, b, c, d, e>: {a, b, c}, {a, b, d}, {a, b, e}, {a, c, d}, {a, c, e}, {a, d, e}, {b, c, d}, {b, c, e}, {b, d, e}, {c, d, e}
- C_2. <a, b, f, g>: {a, b, f}, {a, b, g}, {a, f, g}, {b, f, g}

Jaccard coefficient may lead to wrong clustering result

- C_1: 0.2 ({a, b, c}, {b, d, e}} to 0.5 ({a, b, c}, {a, b, d})
- C_1 & C_2: could be as high as 0.5 ({a, b, c}, {a, b, f})

Jaccard coefficient-based similarity function:

$$Sim(T_1, T_2) = \frac{|T_1 \cap T_2|}{|T_1 \cup T_2|}$$

- Ex. Let T_1 = {a, b, c}, T_2 = {c, d, e}

$$Sim(T_1, T_2) = \frac{|\{c\}|}{|\{a, b, c, d, e\}|} = \frac{1}{5} = 0.2$$

Example on whiteboard

Similarity Measure in ROCK

- Measure based on ‘links’.

- Neighbor: p1 and p2 are neighbors

  iff sim(p1,p2) >= t

  (sim and t between 0 and 1)

- Link(pi,pj) is the number of common neighbors between pi and pj

Similarity Measure in ROCK

- Links: # of common neighbors
  - C_1 <a, b, c, d, e>: {a, b, c}, {a, b, d}, {a, b, e}, {a, c, d}, {a, c, e}, {a, d, e}, {b, c, d}, {b, c, e}, {b, d, e}, {c, d, e}
  - C_2 <a, b, f, g>: {a, b, f}, {a, b, g}, {a, f, g}, {b, f, g}
- Let T_1 = {a, b, c}, T_2 = {c, d, e}, T_3 = {a, b, f}

  and sim the Jaccard coefficient similarity and t=0.5
  - link(T_1, T_2) = 4, *since they have 4 common neighbors*
    - {a, c, d}, {a, c, e}, {b, c, d}, {b, c, e}
  - link(T_1, T_3) = 5, *since they have 5 common neighbors*
    - {a, b, d}, {a, b, e}, {a, b, g}, {a, b, c}, {a, b, f}

Example on whiteboard

Similarity Measure in ROCK

- Link(Ci,Cj) = the number of cross links between clusters Ci and Cj
- G(Ci,Cj)

  = goodness measure for merging Ci and Cj

  = Link(Ci,Cj) divided by the expected number of cross links

Computation of goodness measure on whiteboard

Computation of goodness measure on whiteboard

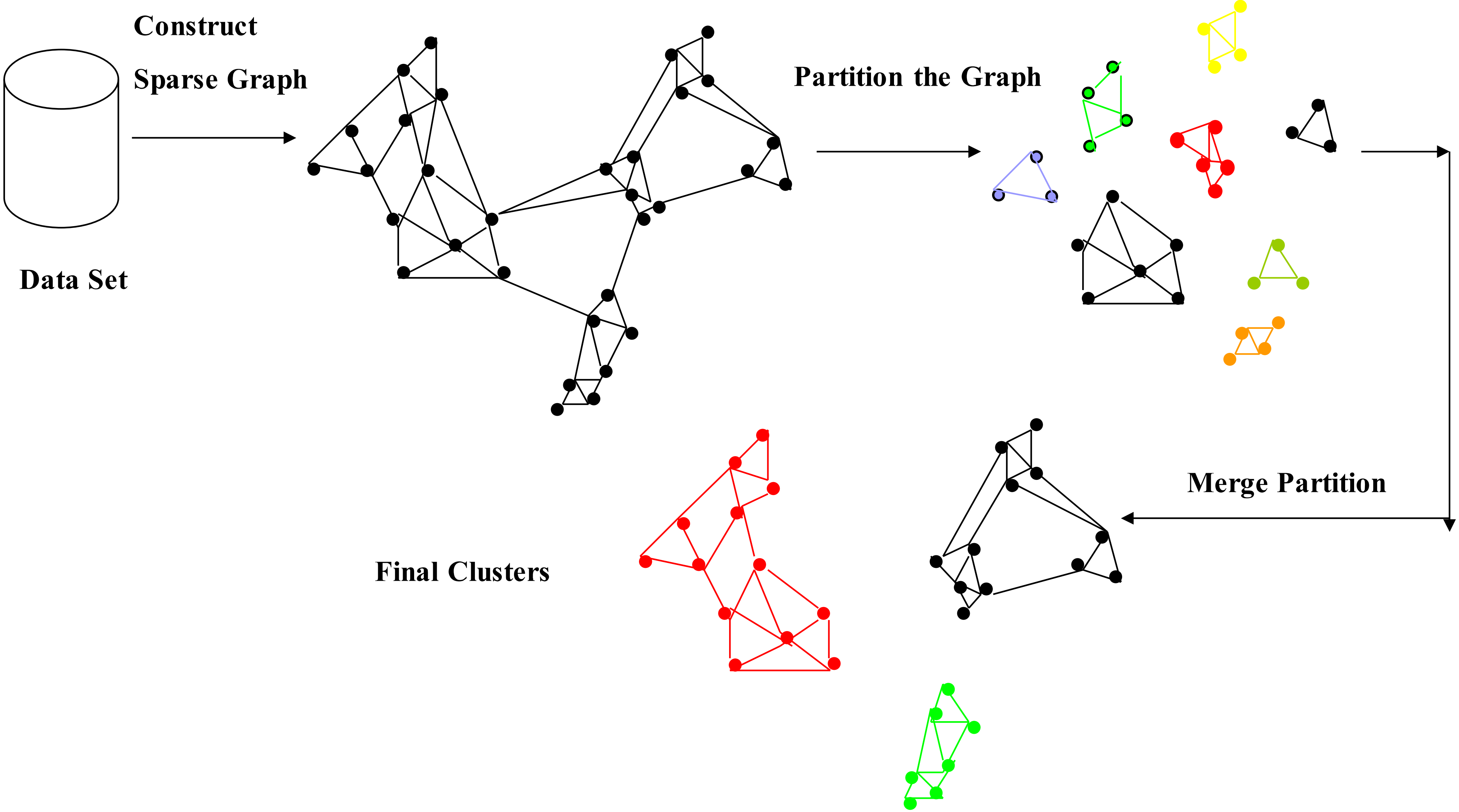
The ROCK Algorithm

- Algorithm: sampling-based clustering
  - Draw random sample
  - Hierarchical clustering with links using goodness measure of merging and desired number of clusters
  - Label data in disk: a point is assigned to the cluster for which it has the most neighbors after normalization

CHAMELEON

- CHAMELEON: by G. Karypis, E.H. Han, and V. Kumar, 1999
- Measures the similarity based on a dynamic model
  - Two clusters are merged only if the *interconnectivity* and *closeness (proximity)* between two clusters are high *relative to* the internal interconnectivity of the clusters and closeness of items within the clusters
- A two-phase algorithm
  1. Use a graph partitioning algorithm: cluster objects into a large number of relatively small sub-clusters
  2. Use an agglomerative hierarchical clustering algorithm: find the genuine clusters by repeatedly combining these sub-clusters

CHAMELEON


Construct
Sparse Graph
Data Set
Partition the Graph
Merge Partition
Final Clusters

CHAMELEON

- A two-phase algorithm

1. Use a graph partitioning algorithm: cluster objects into a large number of relatively small sub-clusters
    - Based on k-nearest neighbor graph
    - Edge between two nodes if points corresponding to either of the nodes are among the k-most similar points of the point corresponding to the other node
    - Edge weight is density of the region
    - Dynamic notion of neighborhood: in regions with high density, a neighborhood radius is small, while in sparse regions the neighborhood radius is large
2.

Example on whiteboard

CHAMELEON

- A two-phase algorithm

1.
2. Use an agglomerative hierarchical clustering algorithm: find the genuine clusters by repeatedly combining these sub-clusters
    - Interconnectivity between clusters Ci and Cj: normalized sum of the weights of the edges that connect nodes in Ci and Cj
    - Closeness of clusters Ci and Cj: average similarity between points in Ci that are connected to points in Cj
    - Merge if both measures are above user-defined thresholds

Explanation on whiteboard

CHAMELEON