Hierarchical Methods

Disadvantages of Classical Hierarchical clustering algorithms

- Lack of robustness (sensitivity to noise and outliers)
- No backtracking (incapable of correcting previous misclassification)
- Computational complexity, which is at least $O(N^2)$
- Difficult in selecting merge or split points
- Hierarchical clustering does not scale well:

Solution : M**ultiple phase clustering** (Combine hierarchical clustering with other clustering techniques)

Recent Advances

- New clustering methods have been designed for clustering a large amount of numeric data by integrating hierarchical clustering (micro stage) and other clustering methods such iterative partitioning (macro stage)
  - **BIRCH** (Balanced Iterative Reducing and Clustering using Hierarchies)
  - **Chameleon:** Multiphase Hierarchical clustering with dynamic modeling

BIRCH

- BIRCH is designed for clustering a large amount of numerical data
- It overcomes the two difficulties of agglomerative clustering methods:
  - **scalability**
  - **the inability to undo what was done in the previous step.**

BIRCH

- BIRCH introduces two concepts:
  - **Clustering Feature (CF)**
    - They are used to summarize cluster representations.
  - **Clustering feature tree (CF tree)**
    - It is used to represent a cluster hierarchy.
- These structures help the clustering method achieve good **speed and scalability** in large databases.
- The structures are effective for incremental and dynamic clustering of incoming objects.

BIRCH Algorithm

- **Clustering Feature (CF)**
  - CF is a three-dimensional vector summarizing information about clusters of objects.
  - Given n d-dimensional objects or points in a cluster, $\{\boldsymbol{x}_i\}$, then the CF of the cluster is defined as:

$$\boldsymbol{CF} = \langle \boldsymbol{n}, \boldsymbol{LS}, \boldsymbol{SS} \rangle$$

  - where n is the number of points in the cluster,
  - $\boldsymbol{LS}$ is the linear sum of the n points, i.e.,

$$\sum_{i=1}^{n} x_i$$

  - $\boldsymbol{SS}$ is the square sum of the data points, i.e.,

$$\sum_{i=1}^{n} {x_i}^2$$

BIRCH Algorithm

- Given n d-dimensional data objects or points in a cluster, we can define the centroid $\boldsymbol{x_0}$, radius $\boldsymbol{R}$, and diameter $\boldsymbol{D}$ of the cluster as follows:

$$x_0 = \frac{\sum_{i=1}^{n} \boldsymbol{x}_i}{n} \quad R = \sqrt{\frac{\sum_{i=1}^{n} (\boldsymbol{x}_i - \boldsymbol{x}_0)^2}{n}} \quad D = \sqrt{\frac{\sum_{i=1}^{n} \sum_{j=1}^{n} (\boldsymbol{x}_i - \boldsymbol{x}_j)^2}{n(n-1)}}$$

- Where R is the average distance from member objects to the centroid, and D is the average pairwise distance within a cluster.
- Both R and D reflect the tightness of the cluster around the centroid.

Clustering feature is summary of the statistics for the given cluster

BIRCH Algorithm

- Clustering features are **additive**.
- For example, suppose that we have two disjoint clusters, C_1 and C_2, having the clustering features, CF_1 and CF_2, respectively.
- The clustering feature for the cluster that is formed by merging C_1 and C_2 is simply $CF_1 + CF_2$.
- Clustering features are sufficient for calculating all of the measurements that are needed for making clustering decisions in BIRCH.

BIRCH Algorithm

- **Example: Clustering feature.**
  - Suppose that there are three points, (2, 5), (3, 2), and (4, 3), in a cluster, C_1. The clustering feature of C_1 is:

$$CF_1 = \langle 3, (2+3+4, 5+2+3), (2^2+3^2+4^2, 5^2+2^2+3^2) \rangle$$
$$= \langle 3, (9, 10), (29, 38) \rangle.$$

  - Suppose that C_1 is joint to a second cluster, C_2, where $CF_2 = \langle 3, (35, 36), (417, 440) \rangle$.
  - The clustering feature of a new cluster, C_3, that is formed by merging C_1 and C_2, is derived by adding CF_1 and CF_2. That is:

$$CF_3 = \langle 3+3, (9+35, 10+36), (29+417, 38+440) \rangle$$
$$= \langle 6, (44, 46), (446, 478) \rangle.$$

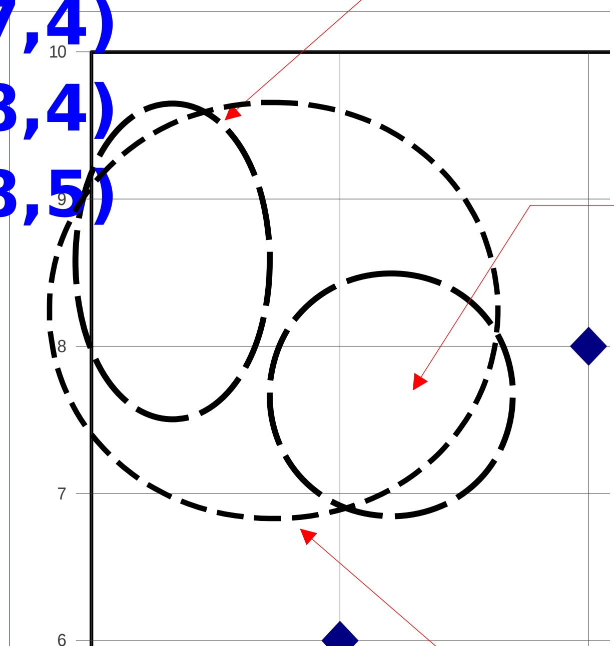
Example of Clustering Feature Vector

- Clustering Feature: $CF = \left(\vec{N}, LS, SS \right)$
- *N*: Number of data points

$$LS : \sum_{i=1}^{N} \vec{X}_i \qquad SS : \sum_{i=1}^{N} \vec{X}_i^2$$

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

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



CF_1 = (5, (16, 30), (54, 190))

CF_2 = (5, (36, 17), (262, 61))

CF = (10, (52, 47), (316, 251))

CF Tree

CF-tree is a height balanced tree that stores clustering features for hierarchical clustering.

CLUSTER FEATURE TREE PARAMETERS:

- **Branching Factor B** : determines the maximum children allowed for a non-leaf node.
- **Threshold T** : T is an upper limit to the radius of a cluster in a leaf node.
- Number of Entries in a **Leaf Node L**

These parameters influence the size of the resulting tree.

- For a CF entry in a root node or a non-leaf node, that CF entry equals the sum of the CF entries in the child nodes of that entry.
- A leaf node CF is referred to simply as a leaf

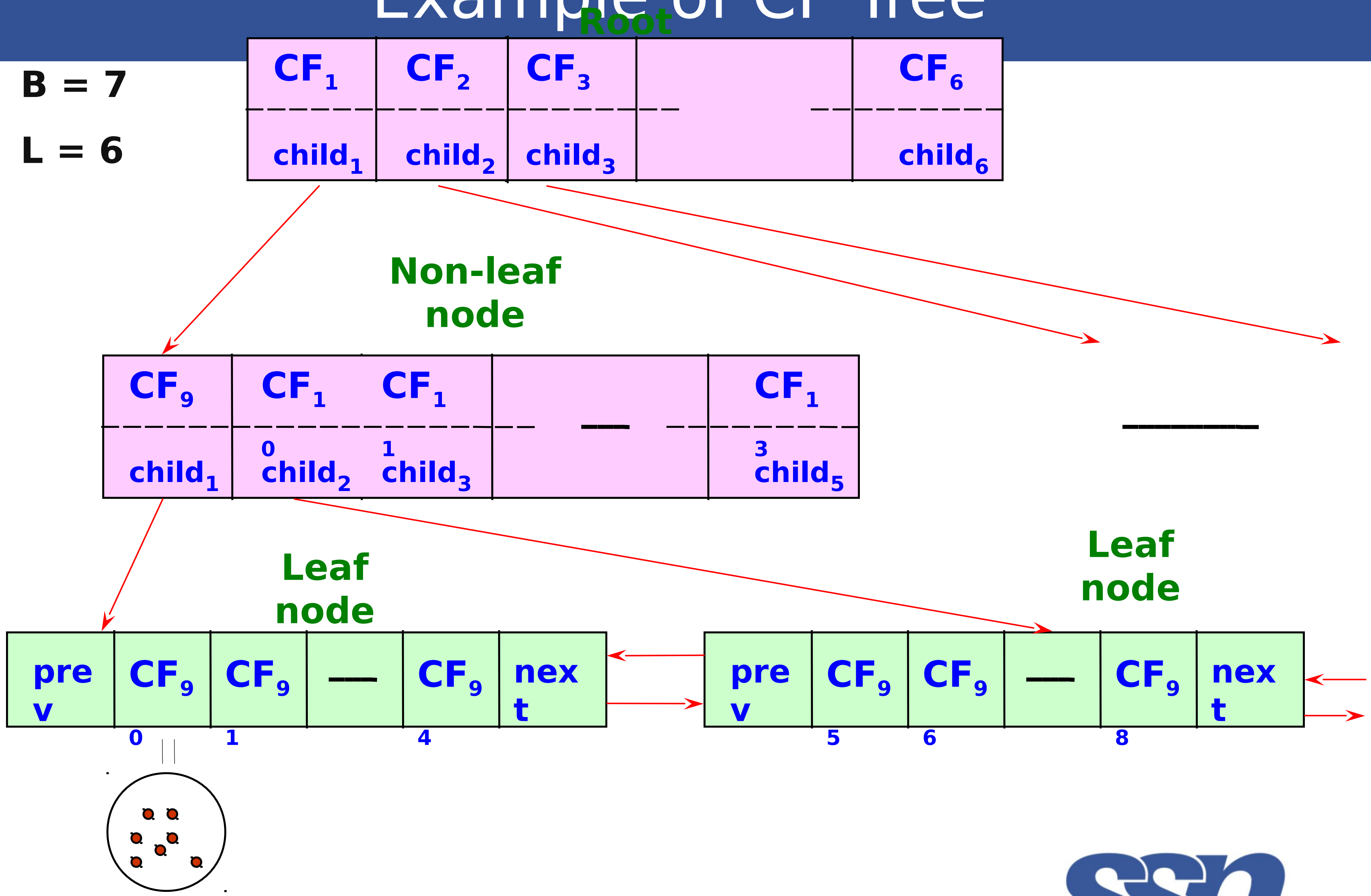
Clustering Feature Tree (CFT)

- Clustering feature tree (CFT) is an alternative representation of data set:
  - Each non-leaf node is a cluster comprising sub-clusters corresponding to entries (at most B) in non-leaf node
  - Each leaf node is a cluster comprising sub-clusters corresponding to entries (at most L) in leaf node
  - Each sub-cluster’s diameter is at most T;
  - Each CF tree should fit in main memory

Example of CF Tree

B = 7

L = 6



BIRCH Algorithm

- BIRCH applies a multiphase clustering technique
  - A single scan of the data set yields good clustering and one or more additional scans can used to improve the quality
- **Phase 1:** BIRCH scans the db to build an initial in-memory CF-tree
  - Preserves the data's inherent clustering technique.
- **Phase 2:** BIRCH applies a (selected) clustering algorithm to cluster the leaf nodes of the CF-tree
  - Removes sparse clusters as outliers and groups dense clusters into large ones.

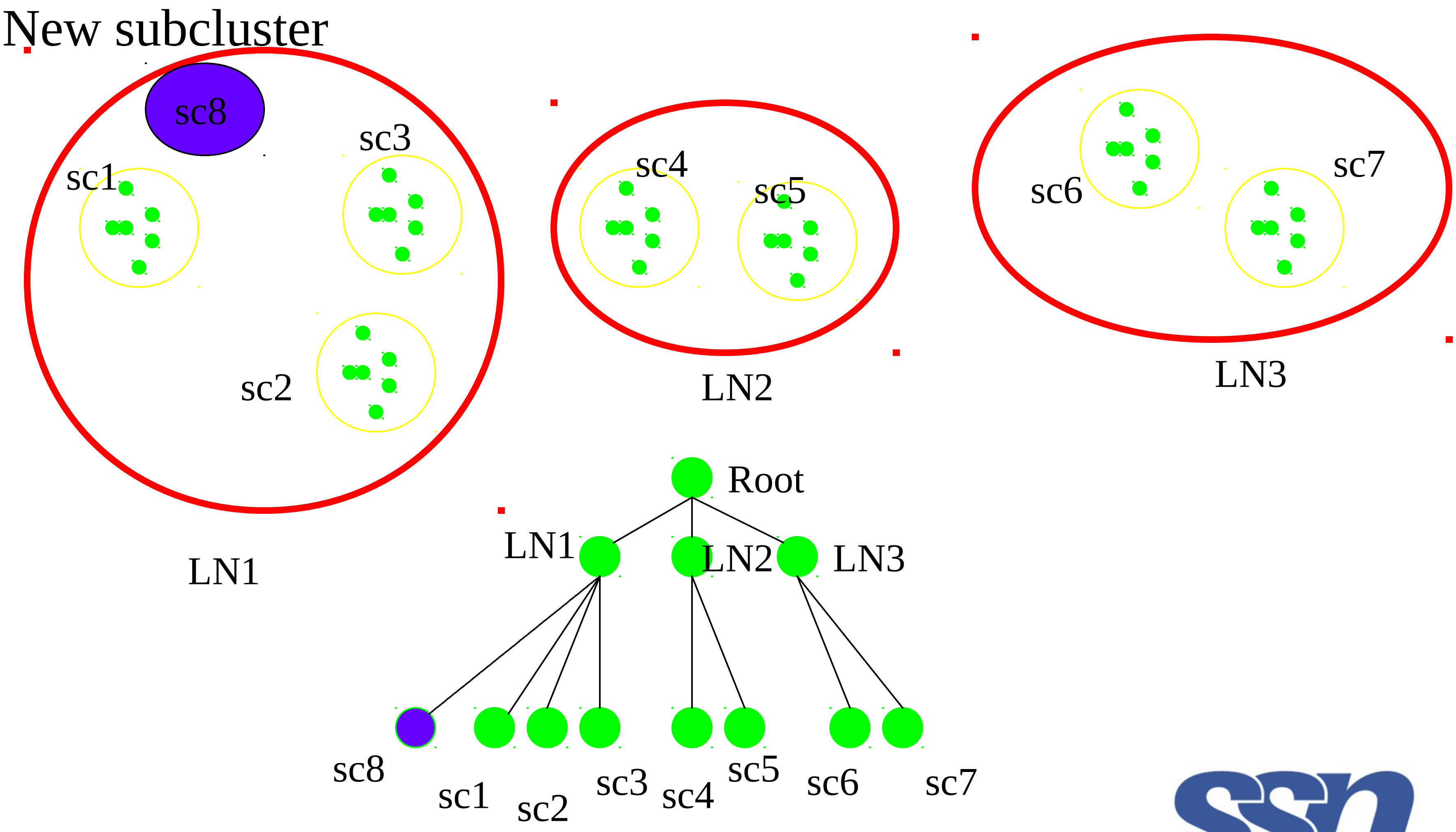
BIRCH Algorithm Phases

- **Phase 1:**
  - the CF tree is built dynamically as objects are inserted.
  - Thus, the method is **incremental**.
  - An object is inserted into the closest leaf entry (subcluster).
  - If the diameterc of the subcluster stored in the leaf node after insertion is larger than the threshold value, then the leaf node and possibly other nodes are split.
  - After the insertion of the new object, information about it is passed toward the root of the tree.
  - The size of the CF tree can be changed by modifying the threshold.

BIRCH Algorithm Phases

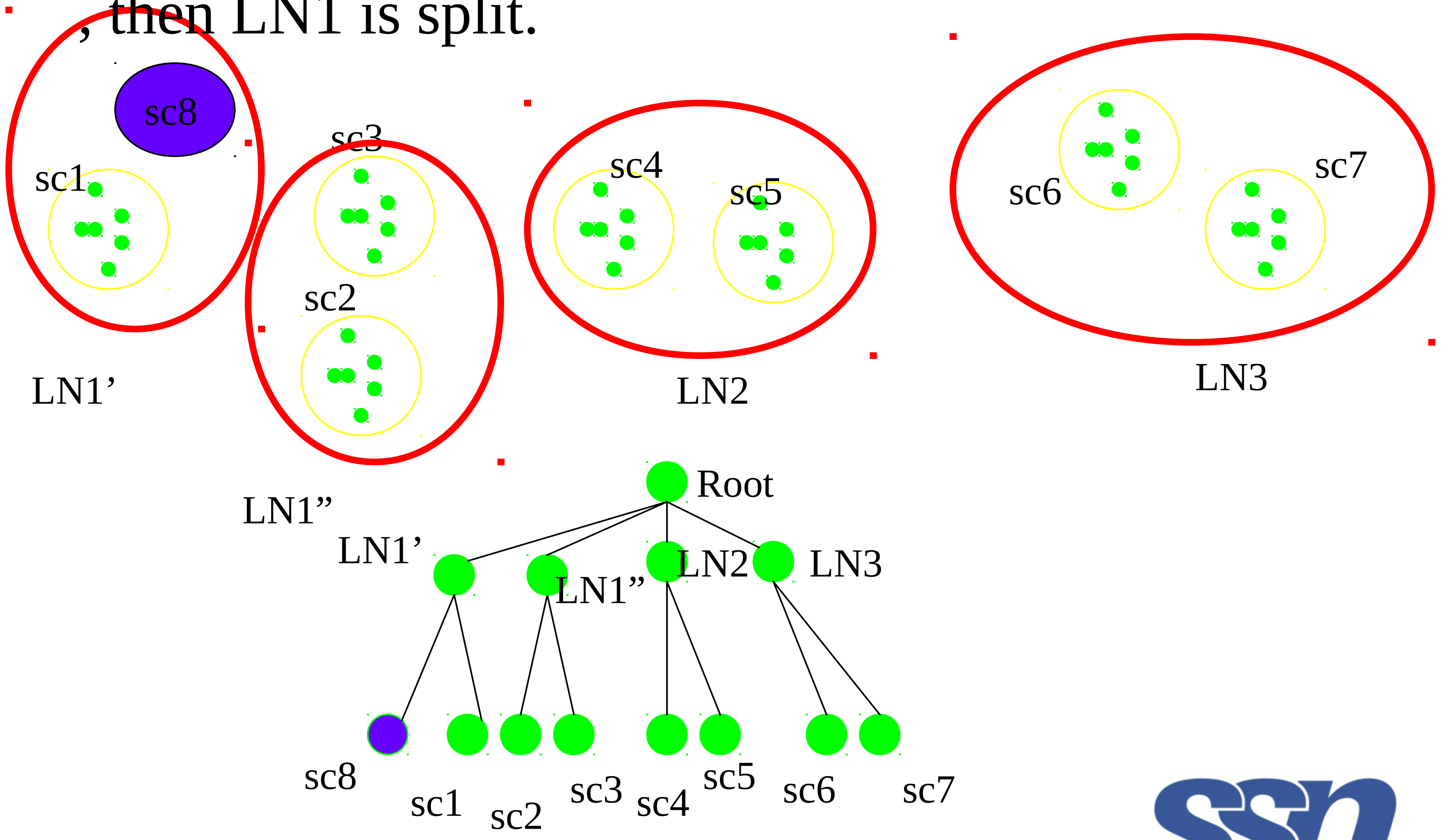
- **Phase 2:**
  - Once the CF tree is built, any clustering algorithm, such as a typical partitioning algorithm, can be used with the CF tree in Phase 2.

Example of the BIRCH Algorithm



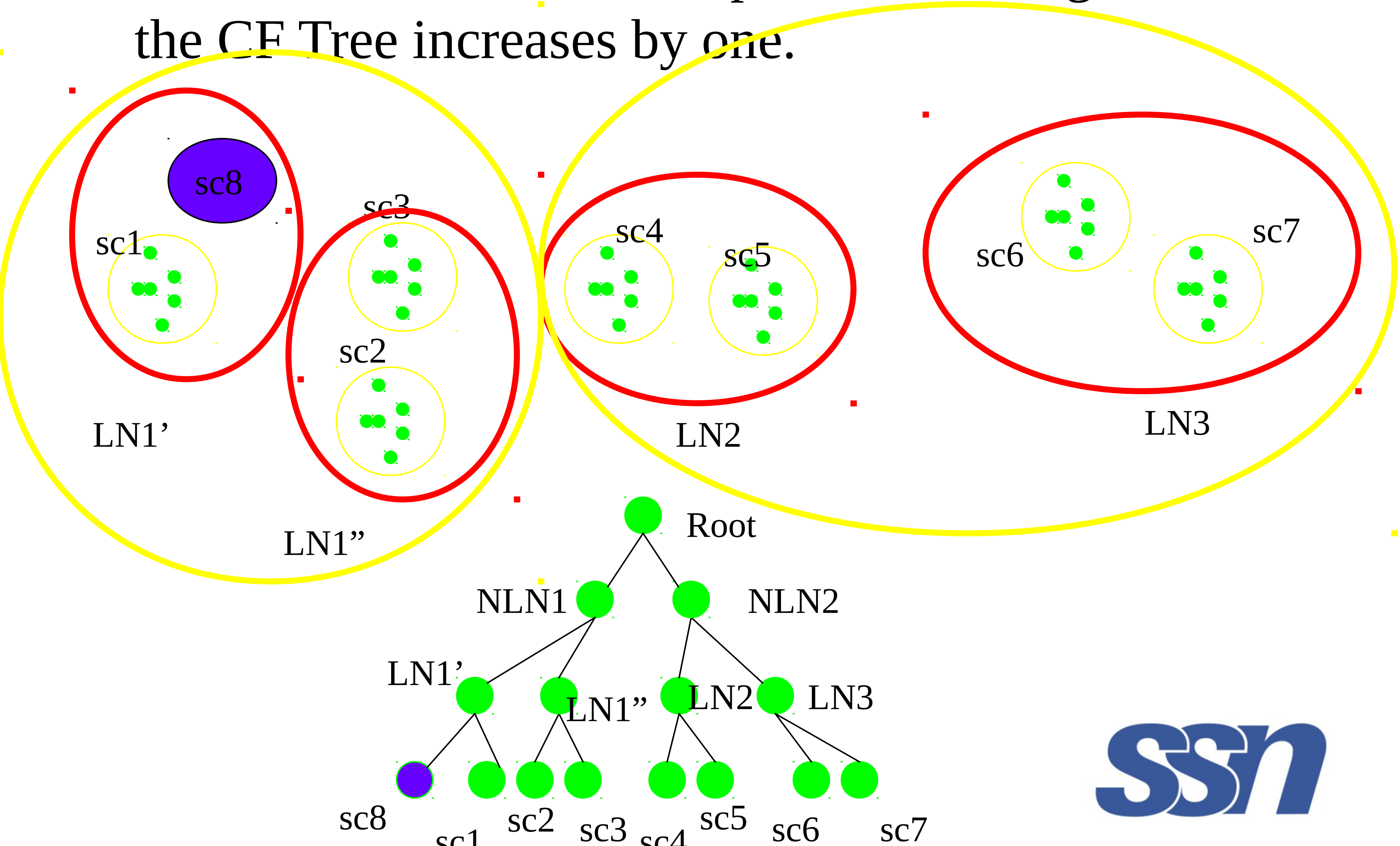
Merge Operation in BIRCH

If the branching factor of a leaf node can not exceed 3, then LN1 is split.



Merge Operation in BIRCH

If the branching factor of a non-leaf node can not exceed 3, then the root is split and the height of the CF Tree increases by one.



Computational Complexity of the Algorithm

- The computation complexity of the algorithm is $O(n)$,
  - were n is the number of objects to be clustered.
- Experiments have shown the linear scalability of the algorithm with respect to the number of objects and good quality of clustering of the data.

Weakness of BIRCH

- However, since each node in a CF tree can hold only a limited number of entries due to its size, a CF tree node does not always correspond to what a user may consider a natural cluster.
- Moreover, if the clusters are not spherical in shape, BIRCH does not perform well, because it uses the notion of radius or diameter to control the boundary of a cluster.

References

- J. Han, M. Kamber, **Data Mining: Concepts and Techniques**, Elsevier Inc. (2006). (Chapter 7)