BIRCH: Balanced Iterative Reducing and Clustering Using Hierarchies

A hierarchical clustering method.

It introduces two concepts :

- Clustering feature
- Clustering feature tree (CF tree)

These structures help the clustering method achieve good speed and scalability in large databases.

1

1

Motivation

- Major weakness of agglomerative clustering methods
  - Do not scale well; time complexity of at least O(n^2), where *n* is total number of objects
  - Can never undo what was done previously
- Birch: Balanced Iterative Reducing and Clustering using Hierarchies
  - Incrementally construct a *CF* (*Clustering Feature*) tree, a hierarchical data structure summarizing cluster info; finds a good clustering with a single scan
  - Apply multi-phase clustering; it improves quality with a few additional scans

2

2

- **Overview of Clustering Algorithms**
  - Clustering algorithms like K-means are inefficient for large datasets.
  - They struggle with limited resources like memory or slower CPUs.
  - As dataset size increases, the running time and clustering quality decrease.
- **BIRCH Clustering**
  - BIRCH stands for Balanced Iterative Reducing and Clustering using Hierarchies.
  - Designed to cluster large datasets efficiently by creating a compact summary.
  - This summary retains as much information as possible and is clustered instead of the larger dataset.
  - BIRCH is often used to complement other clustering algorithms.

- **BIRCH Limitation**
  - BIRCH can only process metric attributes.
  - Metric attributes are values that can be represented in Euclidean space (no categorical attributes).
- **BIRCH Terminology**
  - Clustering Feature (CF): Small, dense regions summarizing large datasets.
  - CF entry: An ordered triple (N, LS, SS) representing a CF.
  - CF Tree: Compact representation containing CF entries.
- **CF Tree Structure**
  - CF tree is a tree where each leaf node contains a sub-cluster.
  - Each entry in a CF tree contains a pointer to a child node and a CF entry.
  - Maximum number of entries in each leaf node is called the threshold.

- **Parameters of BIRCH Algorithm**
  - **Threshold Radius (T): The maximum radius within which points are considered part of the same cluster. Points that fall outside this radius are considered outliers or noise.**
  - **Branching Factor (B): The maximum number of subclusters that can be stored in a non-leaf node of the Clustering Feature Tree (CFT). When a node exceeds this number, it is split into multiple child nodes.**
  - **Maximum Number of Clusters (K): The maximum number of clusters to be generated by the algorithm. This parameter can help control the granularity of the clustering.**
  - **Minimum Number of Points (L): The minimum number of points that a subcluster must contain to be considered a valid cluster. This parameter helps filter out small, insignificant clusters.**
  - **Clustering Criterion (C): The criterion used to determine when to merge subclusters. This can be based on distance (e.g., Euclidean distance) or other similarity measures.**
  - **Memory Size (M): The maximum number of Clustering Feature Tree (CFT) entries that can be stored in memory before a new CFT node is created.**

Summarized Info for Single cluster

- **Given a cluster with N objects**
  - **Centroid**

$$\overrightarrow{X_0} = \frac{\sum_{i=1}^{N} \overrightarrow{X_i}}{N}$$

  - **Radius**

$$R = \left(\frac{\sum_{i=1}^{N} (\overrightarrow{X_i} - \overrightarrow{X_0})^2}{N}\right)^{1/2}$$

  -
  - **Diameter**

$$D = \left(\frac{\sum_{i=1}^{N} \sum_{j=1}^{N} (\overrightarrow{X_i} - \overrightarrow{X_j})^2}{N(N-1)}\right)^{1/2}$$

Summarized Info for Two Clusters

- Given two clusters with N1 and N2 objects, respectively
  - Centroid Euclidean distance

$$D_0 = ((\vec{X}_0 - \vec{Y}_0)^2)^{1/2}$$

  - Centroid Manhattan distance

$$D_1 = \left| \vec{X}_0 - \vec{Y}_0 \right|$$

  - Average inter-cluster distance

$$D_2 = \left(\frac{\sum_{i=1}^{N1} \sum_{j=1}^{N2} (\vec{X}i - \vec{Y}j)^2}{N1N2}\right)^{1/2}$$

7

7

Clustering Feature (CF)

- CF = (*N*, *LS*, *SS*)
  - ***N*** =|C|

    Number of data points
  - ***LS*** = $\sum_{i=1}^{N} \overrightarrow{X_i}$

    Linear sum of N data points
  - ***SS*** =

    Suare sum of N data points

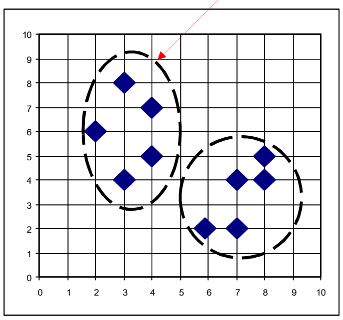
$$\sum_{i=1}^{N} \overrightarrow{X_i}^2$$

8

Example of Clustering Feature Vector

- Clustering Feature: $CF = (\vec{N}, LS, SS)$
- *N*: Number of data points $SS: \sum_{i=1}^{N} \vec{X}_i^2$ $LS: \sum_{i=1}^{N} \vec{X}_i$

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



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

$$R = \left(\frac{\sum_{i=1}^{N}(\overrightarrow{X_i} - \overrightarrow{X_0})^2}{N}\right)^{1/2}$$

$$= \left(\frac{\sum_{i=1}^{N}(\overrightarrow{X_i}^2 - 2\overrightarrow{X_0}\overrightarrow{X_i} + \overrightarrow{X_0}^2)}{N}\right)^{1/2}$$

$$= \left(\frac{\sum_{i=1}^{N}\overrightarrow{X_i}^2 - 2\sum_{i=1}^{N}\overrightarrow{X_i}\overrightarrow{X_0} + \sum_{i=1}^{N}\overrightarrow{X_0}^2)}{N}\right)^{1/2}$$

$$= \left(\frac{\sum_{i=1}^{N}\overrightarrow{X_i}^2 - 2\overrightarrow{X_0}\sum_{i=1}^{N}\overrightarrow{X_i} + N\overrightarrow{X_0}^2}{N}\right)^{1/2}$$

$$= \left(\frac{SS - 2\frac{LS}{N}\cdot LS + N(\frac{LS}{N})^2}{N}\right)^{1/2}$$

$$\boxed{R = \sqrt{\frac{SS - \frac{LS^2}{N}}{N}}}$$

CF Additive Theorem

- Suppose cluster C_1 has $CF_1=(N_1, LS_1, SS_1)$, cluster C_2 has $CF_2=(N_2, LS_2, SS_2)$
- If we merge C_1 with C_2, the CF for the merged cluster C is

$$CF = CF_1 + CF_2$$
$$= (N_1 + N_2, \overrightarrow{LS_1} + \overrightarrow{LS_2}, SS_1 + SS_2)$$

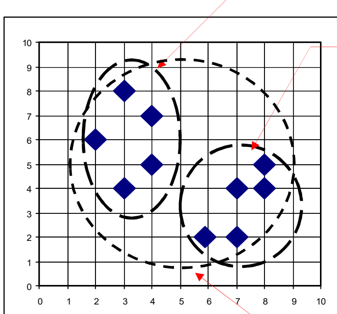
- Why CF?
  - Summarized info for single cluster
  - Summarized info for two clusters
  - Additive theorem

11

11

Example of Clustering Feature Vector

- Clustering Feature: $CF = (\vec{N}, LS, SS)$
- *N*: Number of data points $LS: \sum_{i=1}^{N} \vec{X}_i \quad SS: \sum_{i=1}^{N} \vec{X}_i^2$



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

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

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

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

12

12

Several Issues with the CF Tree

- **Number of entries in CFT node limited by page size; thus, node may not correspond to natural cluster**
  - **Two sub-clusters that should be in one cluster are splitted across nodes**
  - **Two sub-clusters that should not be in one cluster are kept in same node (dependent on input order and skewness)**
- **Sensitive to skewed input order**
  - **Data point may end in leaf node where it should not have been**
  - **If data point is inserted twice at different times, may end up in two copies at two distinct leaf nodes**

13

13