BIRCH Clustering

Rajat Srivastava

- Clustering algorithms like K-means clustering do not perform clustering very efficiently and it is difficult to process large datasets with a limited amount of resources (like memory or a slower CPU). So, regular clustering algorithms do not scale well in terms of running time and quality as the size of the dataset increases. This is where **BIRCH** clustering comes in.

- **Balanced Iterative Reducing and Clustering using Hierarchies (BIRCH**) is a clustering algorithm that can cluster large datasets by first generating a small and compact summary of the large dataset that retains as much information as possible. This smaller summary is then clustered instead of clustering the larger dataset.

- BIRCH is often used to complement other clustering algorithms by creating a summary of the dataset that the other clustering algorithm can now use. However, BIRCH has one major drawback – **it can only process metric attributes. A metric attribute is any attribute whose values can be represented in Euclidean space i.e., no categorical attributes should be present.**

Clustering Feature (CF)
Formally, a Clustering Feature entry is defined as an ordered triple, (N, LS, SS) where ‘N’ is the number of data points in the cluster, ‘LS’ is the linear sum of the data points and ‘SS’ is the squared sum of the data points in the cluster. It is possible for a CF entry to be composed of other CF entries.

Tree Clustering Feature:
The CF tree is the actual compact representation that we have been speaking of so far. A CF tree is a tree where each leaf node contains a sub-cluster.

Threshold
Every entry in a CF tree contains a pointer to a child node and a CF entry made up of the sum of CF entries in the child nodes. There is a maximum number of entries in each leaf node. This maximum number is called the threshold.

Parameters of BIRCH Algorithm :

threshold : threshold is the maximum number of data points a sub-cluster in the leaf node of the CF tree can hold.
branching_factor : This parameter specifies the maximum number of CF sub-clusters in each node (internal node).
n_clusters : The number of clusters to be returned after the entire BIRCH algorithm is complete i.e., number of clusters after the final clustering step. If set to None, the final clustering step is not performed and intermediate clusters are returned.

Algorithms of the BIRCH.

- Without going into the mathematics of BIRCH, more formally, BIRCH is a clustering algorithm that clusters the dataset first in small summaries, then after small summaries get clustered. It does not directly cluster the dataset. This is why BIRCH is often used with other clustering algorithms; after making the summary, the summary can also be clustered by other clustering algorithms.

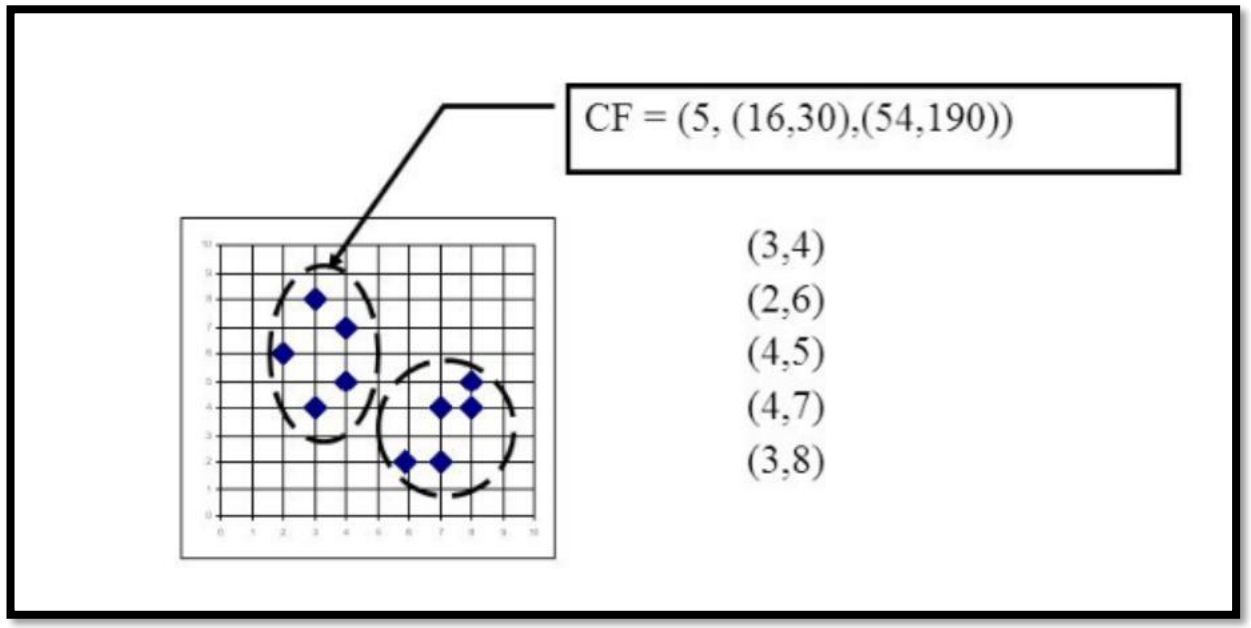
- It is provided as an alternative to MinibatchKMeans. It converts data to a tree data structure with the centroids being read off the leaf. And these centroids can be the final cluster centroid or the input for other cluster algorithms like AgglomerativeClustering.

- BIRCH is a scalable clustering method based on hierarchy clustering and only requires a one-time scan of the dataset, making it fast for working with large datasets. This algorithm is based on the CF (clustering features) tree. In addition, this algorithm uses a tree-structured summary to create clusters. The tree structure of the given data is built by the BIRCH algorithm called the Clustering feature tree(CF tree).

- In context to the CF tree, the algorithm compresses the data into the sets of CF nodes. Those nodes that have several sub-clusters can be called CF subclusters. These CF subclusters are situated in no-terminal CF nodes.

- The CF tree is a height-balanced tree that gathers and manages clustering features and holds necessary information of given data for further hierarchical clustering. This prevents the need to work with whole data given as input. The tree cluster of data points as CF is represented by three numbers (N, LS, SS).

N = Number of items in subclusters
LS = vector sum of the data points
SS = Sum of the squared data points
In the image below, we can easily understand the numbers.



Here we can see in the example how a CF generates there. In the image, five samples are available (3,4), (2, 6), (4, 5), (4, 7), (3, 8),.

So by the image N = 5, LS = (16,30) and SS = (54, 190).

The below image can represent the CF tree structure.

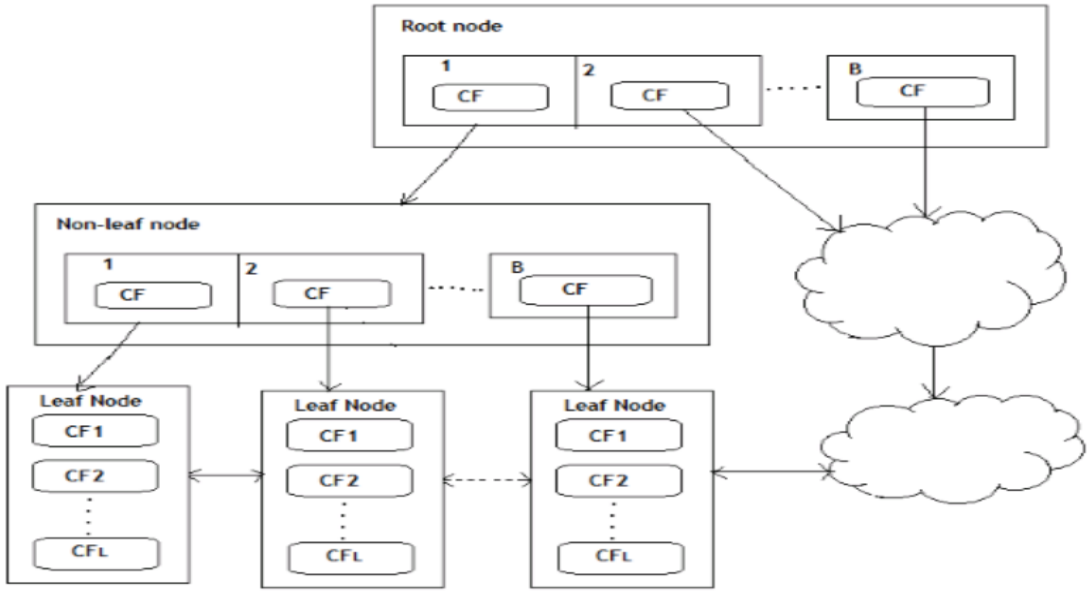


Fig: A CF tree structure

In the image, we can see that the root node has a non-leaf node where every non-leaf node is having B entries and the leaf node has L cluster feature(CF), so if every CF in the leaf node has L entries, it will satisfy the threshold value T is a maximum diameter of radius. So here, we can see that each leaf node is a sub-cluster, more formally saying it is a summary, not a data point.

There are mainly four phases which are followed by the algorithm of BIRCH.

- Scanning data into memory.
- Condense data(resize data).
- Global clustering.
- Refining clusters.

In these four phases, two of them (resize data and refining clusters) are optional. They come in the process when more clarity is required. But scanning data is just like loading data into a model. After loading the data, the algorithm scans the whole data and fits them into the CF trees. In condensing, it resets and resizes the data for better fitting into the CF tree. In global clustering, it sends CF trees for clustering using existing clustering algorithms. Finally, refining fixes the problem of CF trees where the same valued points are assigned to different leaf nodes.