Implementation of Balanced iterative reducing and clustering using hierarchies (BIRCH) clustering algorithm

Ranjan Gsk

Bits Pilani, Hyderabad, India

1.f20160245@hyderabad.bits-pilani.ac.in

***Abstract*— This paper focuses on to implement from scratch and apply Balanced iterative reducing and clustering using hierarchies (BIRCH) clustering algorithm to benchmark artificial datasets (use birch, cure, disk, donut and square datasets).**

Keywords—python , pre-processing , Birch, scikit,matplotlib,clustering.

I. INTRODUCTION

BIRCH stands for ***Balanced Iterative Reducing and Clustering Using Hierarchies***, which uses hierarchical methods to cluster and reduce data. BIRCH only needs to scan the data set in a single pass to perform clustering. Clustering is one of the most widely recognized exploratory information examination procedure used to get an instinct about the structure of the information. It very well may be characterized as the errand of distinguishing subgroups in the information with the end goal that information focuses in a similar subgroup (bunch) are fundamentally the same as while information focuses in various groups are altogether different.

As such, we attempt to discover homogeneous subgroups inside the information with the end goal that information focuses in each bunch are as comparable as conceivable as indicated by a similitude measure, for example, euclidean-based separation or relationship based separation. The choice of which closeness measure to utilize is application-explicit. Clustering investigation should be possible based on highlights where we attempt to discover subgroups of tests dependent on highlights or based on tests where we attempt to discover subgroups of highlights dependent on tests.

Clustering is utilized in showcase division; where we attempt to fined clients that are like each other whether as far as practices or traits, picture division/pressure; where we attempt to assemble comparable districts, report grouping dependent on themes, and so on. In contrast to regulated picking up, clustering is viewed as a solo learning technique since we don't have the ground truth to analyze the yield of the bunching calculation to the genuine marks to assess its exhibition.

II. UNDERSTANDING DATA

The artificial datasets with graph points are used as part of this paper. This dataset has been taken from the clustering benchmark repository in git. The datasets chosen for this paper are

- Birch
- Cure
- Disk
- Donut
- Square

III. PRE-PROCESSING

Before we begin to work with our data we need to pre-process it to get accurate results.Data preprocessing is an important step in the data mining process. The phrase "garbage in, garbage out" is particularly applicable to data mining and machine learning projects. Data-gathering methods are often loosely controlled, resulting in out-of-range values, impossible data combinations. Analyzing data that has not been carefully screened for such problems can produce misleading results. Thus, the representation and quality of data is first and foremost before running an analysis.

The datasets are present in raw format in the github repository and can be bought to the local working directory using terminal commands like curl, wget etc. The datasets are present in .arff formant and can be imported to python using arff module in scipy.io package.

All the points in each dataset were plotted to see how the points were distributed on the two dimensional plain.

IV. UNDERSTANDING BIRCH

The BIRCH algorithm uses a tree structure to create a cluster. It is generally called the Clustering Feature Tree (CF Tree). Each node of this tree is composed of several Clustering features (CF). Each node including leaf nodes has several CFs, and the CFs of internal nodes have pointers to child nodes, and all leaf nodes are linked by a doubly linked list.

Fig.2

In the clustering feature tree, a clustering feature (CF) is defined as follows:Each CF is a triplet, which can be represented by (N, LS, SS).

- Where N represents the number of sample points in the CF, which is easy to understand
- LS represents the vector sum of the feature dimensions of the sample points in the CF
- SS represents the square of the feature dimensions of the sample points in the CF.

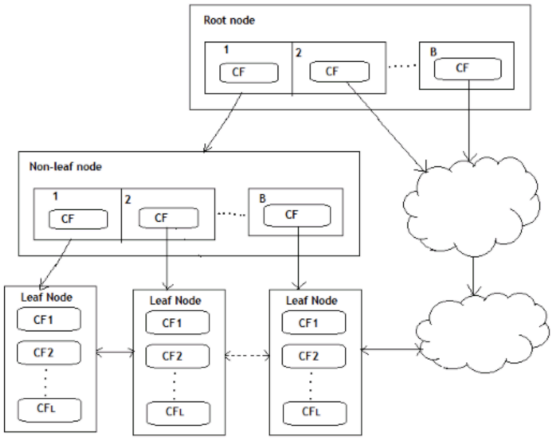
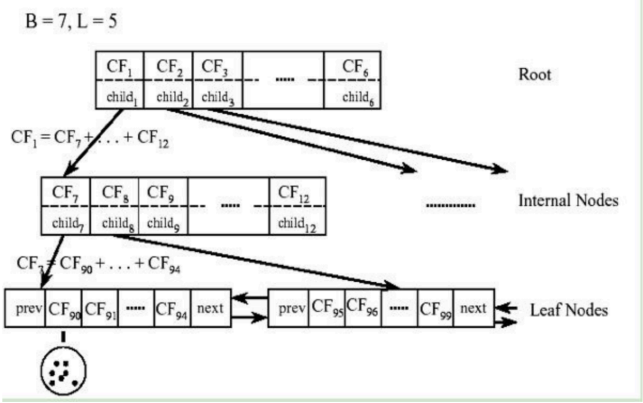


Fig.1. Clustering feature tree

CF has a property. It satisfies the linear relationship, that is:

$$CF_1+CF_2 = (N_1+N_2, LS_1+LS_2, SS_1+SS_2)$$

This property is also well understood by definition. If you put this property on the CF Tree, that is to say, in the CF Tree, for each CF node in the parent node, its (N, LS, SS) triplet value is equal to the CF node pointed to The sum of the triples of all child nodes.



As can be seen from the figure 2, the value of the triplet of CF1 of the root node can be obtained by adding the values of the 6 child nodes (CF7-CF12) that it points to. In this way, we can be very efficient when updating the CF Tree.For CF Tree, we generally have several important parameters,

- The first parameter is the maximum CF number B of each internal node,
- The second parameter is the maximum CF number L of each leaf node,
- The third parameter is for the sample points in a CF in the leaf node. It is the maximum sample radius threshold T of each CF in the leaf node. That is to say, all sample points in this CF must be in the radius In a hyper-sphere less than T.

For the CF Tree in the above figure, B = 7 and L = 5 are defined, which means that the internal node has a maximum of 7 CFs, and the leaf node has a maximum of 5 CFs.

V. GENERATION OF CLUSTERING FEATURE TREE

In the beginning, the CF Tree is empty and there are no samples. The first sample point from the training set is read and put it into a new CF triplet A. That's why N1 initially.

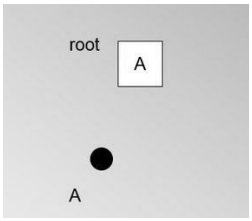


Fig.3

Now we continue to read the second sample point, we find that this sample point and the first sample point A are within the range of a hyper-sphere with a radius T. Now, N=2 in the triplet of A.

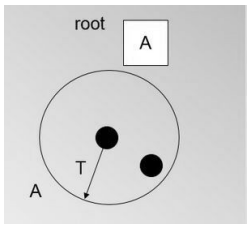


Fig.4

At this point, the third node came, and we found that this node could not be integrated into the hyper-sphere formed by the previous node, that is, we needed a new CF triple B to

accommodate this new value.At this time, the root node has two CF triples A and B. The CF Tree is as follows:

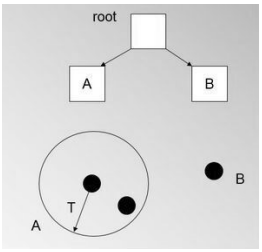


Fig.5

When we came to the fourth sample point, we found that the radius of B and B is smaller than T

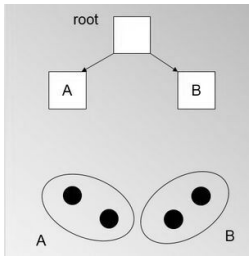


Fig.6

The leaf node LN1 has three CFs, and LN2 and LN3 each have two CFs. The maximum CF number of our leaf nodes is L = 3. At this time a new sample point is coming, we find that it is closest to the LN1 node, so we start to judge whether it is within the super sphere corresponding to the three CFs of sc1, sc2, sc3. The problem is that our L = 3, which means that the number of CFs of LN1 has reached the maximum value, and no new CF can be created

If the maximum CF number of our internal nodes is B = 3, then splitting the leaf node into two will cause the maximum CF number of the root node to be exceeded, that is to say, our root node will now also split, the split method and The leaf nodes are split, and the split CF Tree is as follows:

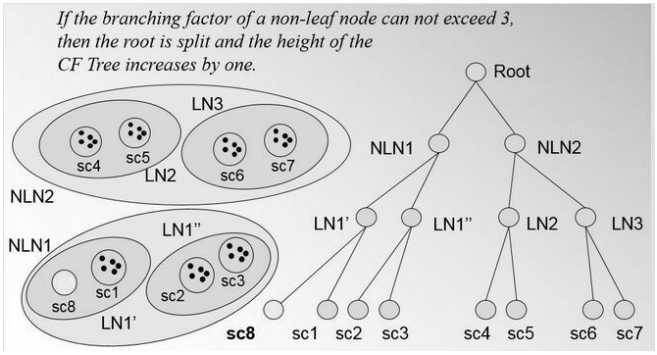


Fig.7

Summary of the insertion of CF Tree

1. Find the **leaf node** closest to the new sample and the **closest CF node** in the leaf node from the root node

2. After the new sample is added, if the radius of the hyper-sphere corresponding to this CF node still satisfies the threshold T, **then all the CF triplets on the path are updated, and the insertion ends.** Otherwise, go to 3.

3. If the number of CF nodes of the current leaf node is less than the threshold L, create a new CF node, put in a new sample and the new CF node into this leaf node, update all CF triplets on the path, and insertion Ends. Otherwise, go to 4

4. Divide the current leaf node into two new leaf nodes, select the two CF tuples with the farthest hyper-sphere among all CF tuples in the old leaf node, and distribute as the first CF node of the two new leaf nodes. Put other tuples and new sample tuples into corresponding leaf nodes according to the principle of distance.

5. In turn, check whether the parent node is also to be split. If it needs to be split in the same way as the leaf node.

VI. BIRCH ALGORITHM

All the training set samples are built into a CF Tree, and a basic BIRCH algorithm is completed. The corresponding output is several CF nodes, and the **sample point in each node is a cluster.**In addition to building a CF Tree to cluster the real BIRCH algorithm, there are some optional algorithm steps. Now let's take a look at the process of the BIRCH algorithm.

- Read all the samples in sequence and create a CF Tree in memory. For the method of creation, refer to the previous section.

- (Optional) Filter the CF Tree created in the first step to remove some abnormal CF nodes.

The main purpose of this step is to eliminate problems caused by some tree structure splits **due to the limitation of the number of CF nodes**.

- (Optional) Using the centroids of all CF nodes of the CF Tree generated in the third step as initial centroid points, cluster all the sample points according to distance

Summary

- The BIRCH algorithm **does not need to input the K value of the number of categories**, which is different from K-Means and Mini Batch K-Means.

- If you do not enter the K value, the number of the last CF tuple is the final K, otherwise, the CF tuple will be merged according to the distance

- In addition to clustering, BIRCH can also do some additional outlier detection and preliminary data pre-processing according to category specifications.

VII. Conclusion

The scikit library has been used to implement the above Algorithm . The threshold values of the plots have been Modified to get the perfect clusters for each dataset. The birch algorithm takes three inputs which are Branching factor , threshold and number of clusters. Number of clusters has been set to none as it returns The subclusters as they are without performing the final clustering step. Branching factor gives the maximum number of CF clusters in each node.The branching factor has been set to 50 for all datasets.Threshold tells us about minimum radius required to merge a new sample and closest sub cluster.The threshold values for the datasets for accurate clustering are

- Birch :1.5
- Cure : 0.8
- Disk : 0.5
- Donut : 0.1
- Square : 2