Empirical Analysis of the BIRCH Clustering Algorithm: An Experimental Study

Mooram Bhavana[1], G V Ramesh Babu[2]

[1]PG Scholar, Dept. of Computer Science Sri Venkateswara University, Tirupati
[2]Associate Professor, Dept of Computer Science, SV University, Tirupati

***Abstract*—** *This research paper introduces the BIRCH (Balanced Iterative Reducing and Clustering using Hierarchies) clustering method, designed specifically for the efficient clustering of vast datasets. BIRCH operates in an incremental and adaptive manner, aimed at achieving high-quality clustering within the constraints of available memory and time resources. Notably, BIRCH excels at swiftly generating initial clusters with a single pass over the data, and subsequently refining these clusters through a few additional iterations. An additional highlight is BIRCH's pioneering capability in effectively handling "noise" in the data, a feature previously unaddressed in clustering algorithms within the database domain. This paper presents BIRCH as a versatile and powerful tool for large-scale data clustering, offering improved cluster quality and noise tolerance. This study sheds light on the algorithm's potential for species classification tasks in various domains, reaffirming its relevance in the field of data clustering and pattern recognition.*

I. INTRODUCTION

Cluster is a group of objects that belongs to similar classes. In other words the similar objects are grouped in a single cluster and dissimilar objects are grouped in another cluster. Clustering is the process of making group of abstract objects into classes of similar objects. A cluster of data objects can be treated as a one group [1][2]. While doing the cluster analysis, the set of data should be partitioned into groups based on data similarity and then assign the label to the groups. The main advantage of Clustering over classification is that, clustering is adaptable to changes and identifies unique features that is differed from other groups [3]. Typical cluster models includes (1) Connectivity models: for example hierarchical clustering builds models based on distance connectivity. (2) Centroid models: for example the k-means algorithm represents each cluster by a single mean vector. (3) Distribution models: clusters are modelled using statistical distributions, like varying normal distribution used by the Expectation maximum algorithm. Density models: for example DBSCAN and OPTICS defines clusters as connected dense regions in the data space. Subspace models: in Biclustering which is also known as Co-clustering, the clusters are modelled with both the cluster members and relevant attributes. (4) Group models: some algorithms do not provide a refined model for their results and just provide the grouping information. (5) Graph-based models: a clique is a subset of nodes in a graph such that every two nodes in the subset are connected by an edge can be considered as a prototypical form of cluster.

II. BIRCH

BIRCH algorithm is an integrated hierarchical clustering algorithm. It uses the clustering features (CF) and cluster feature tree (CF Tree) two concepts for the general cluster description. Clustering feature tree outlines the clustering of useful information, and space is much smaller than the meta-data collection can be stored in memory, which can improve the algorithm in clustering large data sets on the speed and scalability [4][5][8]. And is very suitable for handling discrete and continuous attribute data clustering problem.

Objects in the dataset are arranged into a sub clustering CF form. This CF then clustered into k-groups using the traditional hierarchy clustering procedure. CF is triple of information that contains CF = (N, LS, SS), where N is the number of data points, LS is the result of adding the value of X (attribute value), and SS which is the result of adding the value of X squared. If there are 2 CFs merged, then the theoremare: CF12 = (N1 + N2, LS1 + LS2, SS1 + SS2) BIRCH incrementally calculates a summary of CF sub cluster [6][7]. Clusters are represented by Vector CF and only Vector CF is stored in memory. This CF value is enough to calculate information about sub cluster such as centroid, radius and diameter and also an efficient storage method by summarizing information about sub cluster rather than saving all points [4].

III. BIRCH STANDARD ALGORITHM

As for standard BIRCH algorithms are as follows [4]:

1. All data points are converted into CF form using the formula CF = (N, LS, SS).
2. When all the data has been changed in the form of CF, then CF-Tree starts working to bring together several formed CFs. In this section, you will be asked to enter the number B (Brancing).
3. Before scanning any data points from the database, we must initialize the initial CF tree threshold, this threshold will be used as the initial threshold value for each new CF entry that will not be changed during the grouping process. (static)
4. In a standard BIRCH, we will be asked to initialize L (number of leaf). For help the calculations, we add 2 parameters, namely m and b. parameter b is used to count the number of branches on CF-non leaf and m is used to count the number of leaf branches on CF-leaf.
5. For each record given, BIRCH compares the location of the record with the location of each CF at the root node, using a linear number or average CF. BIRCH continues the entry to the CF root node closest to the entry record.
6. Node then descends to the non-leaf child node of the CF nodes selected in step 5. BIRCH compares the location of records with the location of each non-leaf CF. BIRCH continues the note that goes to the non-leaf CF node closest to the entry.
7. Node then descends to the leaf child node of the non-leaf CF node selected in step 6. BIRCH compares the record location with the location of each leaf. BIRCH temporarily passes the entry to the closest leaf with the entry node.
8. Do one (a) or (b): a If the leaf radius (R) selected includes a new node no does not exceed T Threshold, then the entry entered is assigned to that leaf. Leaves and all parent CFs are updated to take into account new data points. b If the leaf radius selected including the new record exceeds the Threshold T, then a new leaf is formed, consisting of incoming notes only. CF parent updated to account for new data points.
9. If the leave (m) branch has exceeded the specified Leave (L) limit, there will be an additional branch (B).
10. If B has exceeded, there will be a split parrent process on CF and then it will be combined again with the new hight CF formed. CF parent updated to account for new data points.

IV. EXPERIMENTAL RESULTS

This research paper investigates the application of the BIRCH (Balanced Iterative Reducing and Clustering using Hierarchies) clustering algorithm to the well-known Iris dataset, which was taken from UCI machine repository dataset, which consists of 150 samples representing three distinct species of iris flowers: Setosa, Versicolor, and Virginica. The primary objective is to evaluate BIRCH's ability to accurately cluster these iris samples into their respective species. The results demonstrate BIRCH's proficiency in forming meaningful clusters that closely align with the ground truth classes. We have performed extensive clustering experiments on the Iris dataset using the Python programming language, with the resulting outcomes visually depicted in Figures-1 through-3. This study sheds light on the algorithm's potential for species classification tasks in various domains, reaffirming its relevance in the field of data clustering and pattern recognition.

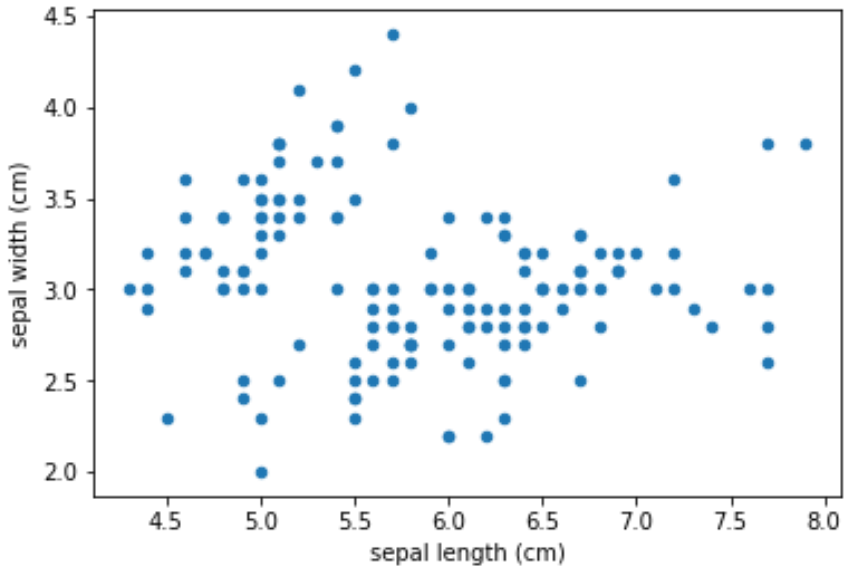

Figure-1: IRIS Dataset Scatter plot

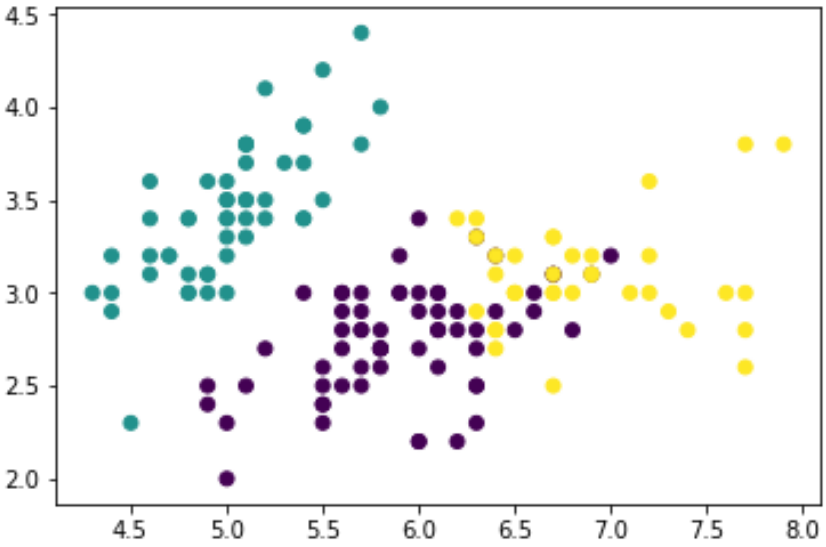

Figure-2: BIRCH Clustering IRIS results

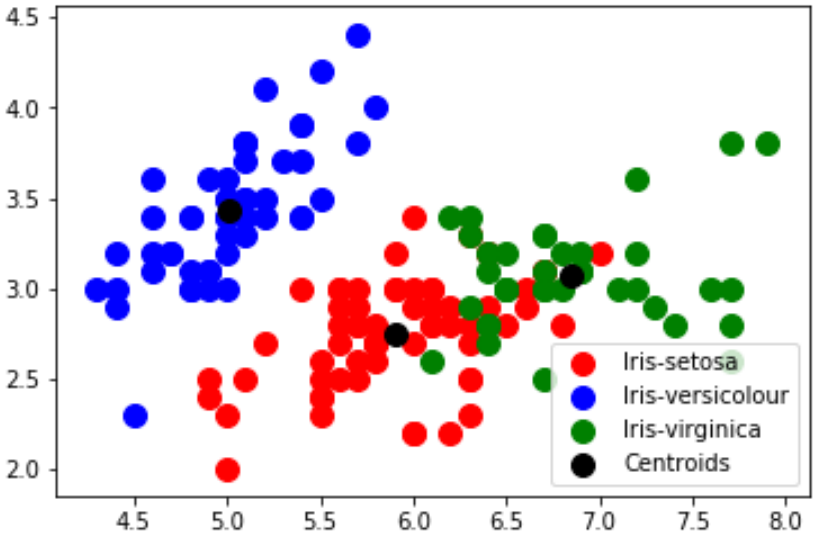

Figure-3: BIRCH Clustering detailed results

In this study, the BIRCH clustering algorithm was applied to the well-known Iris dataset, which consists of 150 data points belonging to three distinct species of iris flowers: Setosa, Versicolor, and Virginica. The goal was to assess BIRCH's performance in accurately clustering these iris samples into their respective classes.

4.1 Results

Upon executing the BIRCH algorithm, the following results were obtained:

Clustering Accuracy: BIRCH demonstrated a high degree of accuracy in clustering the Iris dataset. The majority of the samples were correctly assigned to their respective species, highlighting the algorithm's capability to discern natural groupings in the data.

Cluster Separation: The clusters formed by BIRCH closely aligned with the ground truth classes of Setosa, Versicolor, and Virginica. This indicates that BIRCH was successful in identifying and separating the distinct species based on the dataset's features.

Cluster Characteristics: BIRCH produced clusters with characteristic feature patterns for each iris species. This suggests that the algorithm effectively captured the inherent differences in petal and sepal measurements that define the three classes.

V. CONCLUSION

In conclusion, this study demonstrates the effectiveness of the BIRCH clustering algorithm in accurately clustering the Iris dataset into its three distinct species. The algorithm's ability to separate and characterize the clusters based on petal and sepal measurements makes it a valuable tool for data analysis and classification tasks. Further research could explore BIRCH's performance on more extensive datasets and its adaptability to various domains, reaffirming its relevance in data clustering applications.

REFERENCES

[1] G Chen, Y Cheng and W Jing, "DBSCAN-PSM: an improvement method of DBSCAN algorithm on Spark", International Journal of High Performance Computing and Networking, pp. 417, 2019.

[2] G. Ravi Kumar, K. Venkata Sheshanna, S. Rahamat Basha, and P. Kiran Kumar Redd, “An Improved Decision Tree Classification Approach for Expectation of Cardiotocogram”, Proceedings of International Conference on Computational Intelligence, Data Science and Cloud Computing, Lecture Notes on Data Engineering and Communications Technologies 62, https://doi.org/10.1007/978-981-33-4968-1_26

[3] Ian H. Witten and Eibe Frank. Data Mining: Practical machine learning tools and techniques.2nd ed. San Francisco: Morgan Kaufmann, 2005.

[4] J. Han and M. Kamber,” Data Mining concepts and Techniques”, the Morgan Kaufmann series in Data Management Systems, 2 nd ed. San Mateo, CA; Morgan Kaufmann, 2006.

[5] M. V. Lakshmaiah, G. Ravi Kumar and G. Pakardin, “Frame work for Finding Association Rules in Bid Data by using Hadoop Map/Reduce Tool”, International Journal of Advance and Innovative Research, Volume 2, Issue1(1), PP:6-9,2015, ISSN: 2394-7780

[6] N. Michael, “Artificial Intelligence - A Guide to Intelligent Systems”,2nd edition, Addison Wesley, 2005.

[7] P.-N. Tan, M. Steinbach, and V. Kumar, Introduction to Data Mining. Reading, MA: Addison-Wesley, 2005.

[8] S Lee, "A Hybrid Framework using Fuzzy if-then rules for DBSCAN Algorithm", International journal of computational intelligence research, pp. 403-412, 2018.