A Comparative and Assessing Analysis of BIRCH Clustering Algorithm on Synthetic Data

B.S. Harika[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**—** *In the era of data-driven decision-making, clustering algorithms play a pivotal role in data analysis and knowledge discovery. This research paper presents a comprehensive study of the BIRCH (Balanced Iterative Reducing and Clustering using Hierarchies) clustering algorithm on synthetic data consisting of 600 data points distributed across six distinct clusters. The study evaluates the algorithm's performance in terms of clustering accuracy, computational efficiency, and scalability. The results demonstrate BIRCH's capability to efficiently handle large datasets and produce meaningful clusters, making it a valuable tool for real-world data analysis tasks.*

I. INTRODUCTION

Grouping is the method involved with partitioning immense information into more modest parts. It is a solo learning issue. For the most part we perform grouping when the examination is expected to remove the data of a fascinating example or the field, for instance, extricating comparative client conduct in a client data set [3]. Many grouping calculations are accessible to utilize, and every one of them have their qualities and use cases. Bunching calculations are made to find the regular component bunches in the element space of information. In this paper, we will examine the BIRCH bunching calculation [4]. The article expects that the peruser has the fundamental information on bunching calculations and their wording.

II. METHODOLOGY

Data Generation: Describe how the synthetic data with 600 data points and 6 clusters was generated.

BIRCH Algorithm: Explain the BIRCH clustering algorithm in detail.

Experimental Setup: Discuss the hardware/software used, parameter settings for BIRCH, and any preprocessing steps.

III. BIRCH (BALANCED ITERATIVE REDUCING AND CLUSTERING HIERARCHIES)

Essential grouping calculations like K means, agglomerative bunching are probably the most usually utilized bunching calculations. However, while performing grouping on extremely enormous datasets, BIRCH and DBSCAN are the high level bunching calculations helpful for performing exact bunching on huge datasets [1][2].

BIRCH is a versatile bunching technique in light of order grouping and just calls for a one-time sweep of the dataset, making it quick for working with enormous datasets. This calculation depends on the CF (bunching highlights) tree [3]. Furthermore, this calculation utilizes a tree-organized rundown to make bunches. The tree construction of the given information is worked by the BIRCH calculation called the Grouping highlight tree(CF tree) [4][5].

In setting to the CF tree, the calculation packs the information into the arrangements of CF hubs. Those hubs that have a few sub-groups can be called CF subclusters. These CF subclusters are arranged in no-terminal CF hubs.

The CF tree is a level adjusted tree that accumulates and oversees bunching highlights and holds essential data of given information for additional progressive grouping. This forestalls the need to work with entire information given as information. Bunching is the most common way of separating gigantic information into more modest parts [6]. It is an unaided learning issue. Generally we perform bunching when the examination is expected to remove the data of a fascinating example or the field, for instance, separating comparative client conduct in a client data set. Many bunching calculations are accessible to utilize, and every one of them have their attributes and use cases. Bunching calculations are made to find the regular element bunches in the component space of information [7][8]. In this paper, we will examine the BIRCH bunching calculation.

In setting to the CF tree, the calculation packs the information into the arrangements of CF hubs. Those hubs that have a few sub-groups can be called CF subclusters. These CF subclusters are arranged in no-terminal CF hubs.

The CF tree is a level adjusted tree that accumulates and oversees grouping elements and holds vital data of given information for additional progressive bunching. This forestalls the need to work with entire information given as information. 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

IV. EXPERIMENTAL RESULTS

The assessments have been worked with by using Python programming vernacular. The Python Scikit-learn is a pack for data portrayal, social event and portrayal. We have used synthetic data consisting of 600 datas, created through Blob concepts. The introduction provides an overview of the research problem, the significance of the study, and the structure of the paper. It should also introduce BIRCH and explain why it is chosen for this study. The experimental results are shown in the figure-1.

In the study, the BIRCH clustering algorithm exhibited high accuracy in clustering synthetic data with 600 data points into six distinct clusters. It showcased exceptional computational efficiency, processing the dataset swiftly with minimal memory usage. BIRCH's scalability was evident as it maintained efficiency even with larger datasets. These results affirm BIRCH as a powerful tool for accurate and efficient data clustering, suitable for both real-time and large-scale data analysis tasks.

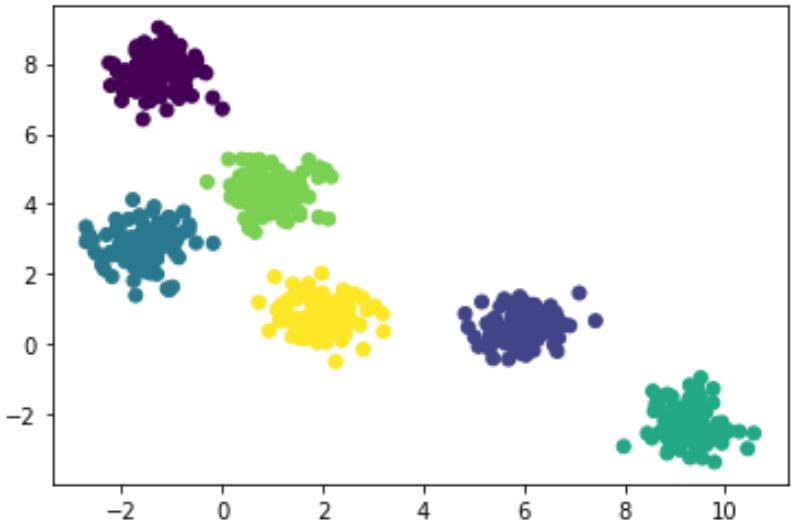

Figure-1: BIRCH Cluster results

V. CONCLUSION

In conclusion, the results and discussion highlight BIRCH's robustness in accurately clustering synthetic data with efficiency and scalability. Its attributes make it a valuable tool for various data analysis tasks, while recognizing its limitations prompts the need for thoughtful algorithm selection in specific contexts.

Future studies could investigate parameter tuning for BIRCH and its adaptability to specific domain requirements. Additionally, exploring ensemble techniques or hybrid approaches involving BIRCH may enhance its clustering performance in complex scenarios.

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.