Comparative Analysis of BIRCH and CURE Hierarchical Clustering Algorithm using WEKA 3.6.9

Yogita Rani*, Manju** & Harish Rohil***

*M.Tech Scholar, Department of Computer Science & Applications, Chaudhary Devi Lal University, Sirsa, Haryana, INDIA.
E-Mail: y.satija2008{at}gmail{dot}com
**Lecturer, Department of Computer Engineering, CDL Government Polytechnic Education Society, Nathusari Chopta, Sirsa, Haryana, INDIA. E-Mail: manju.rohil{at}gmail{dot}com
***Assistant Professor, Department of Computer Science & Applications, Chaudhary Devi Lal University, Sirsa, Haryana, INDIA.
E-Mail: harishrohil{at}gmail{dot}com

***Abstract*—**Hierarchical Clustering is the process of forming a maximal collection of subsets of objects (called clusters), with the property that any two clusters are either disjoint or nested. Hierarchical clustering combine data objects into clusters, those clusters into larger clusters, and so forth, creates a hierarchy of clusters, which may represent a tree structure called a dendrogram, in which the root of the tree consists of a single cluster containing all observations, and the leaves correspond to individual observations. BIRCH and CURE are two integrated hierarchical clustering algorithm. These are not pure hierarchical clustering algorithm, some other clustering algorithms techniques are merged in to hierarchical clustering in order to improve cluster quality and also to perform multiple phase clustering. This paper presents a comparative analysis of these two algorithms namely BIRCH and CURE by applying Weka 3.6.9 data mining tool on Iris Plant dataset.

***Keywords*—**BIRCH Algorithm; CURE Algorithm; Data Mining; Hierarchical Clustering Algorithm; WEKA.

***Abbreviations*—**Balanced Iterative Reducing and Clustering using Hierarchies (BIRCH); Clustering Using REpresentatives (CURE); Waikato Environment for Knowledge Analysis (WEKA).

I. INTRODUCTION

IN data mining hierarchical clustering works by grouping data objects into a tree of cluster. Hierarchical clustering methods can be further classified into agglomerative and divisive hierarchical clustering. This classification depends on whether the hierarchical decomposition is formed in a bottom-up or top-down fashion. Hierarchical techniques produce a nested sequence of partitions, with a single, all inclusive cluster at the top and singleton clusters of individual objects at the bottom. Each intermediate level can be viewed as combining two clusters from the next lower level or splitting a cluster from the next higher level. The result of a hierarchical clustering algorithm can be graphically displayed as tree, called a dendrogram. This tree graphically displays the merging process and the intermediate clusters. This graphical structure shows how points can be merged into a single cluster [Ravichandra Rao, 2003].

Hierarchical methods suffer from the fact that once we have performed either merge or split step, it can never be undone. This inflexibility is useful in that it leads to smaller computation costs by not having to worry about a combinatorial number of different choices. However, such techniques cannot correct mistaken decisions that once have taken. There are two approaches that can help in improving the quality of hierarchical clustering: (1) Firstly to perform careful analysis of object linkages at each hierarchical partitioning or (2) By integrating hierarchical agglomeration and other approaches by first using a hierarchical agglomerative algorithm to group objects into micro-clusters, and then performing macro-clustering on the micro-clusters using another clustering method such as iterative relocation [Jiawei Han & Micheline Kamber, 2006]. One promising direction for improving the clustering quality of hierarchical methods is to integrate hierarchical clustering with other clustering techniques for multiple phase clustering. So in order to make improvement in hierarchical clustering we merge some other techniques or method in to it. This improved version of hierarchical clustering is represented by our two algorithms BIRCH and CURE hierarchical clustering [Almeida et al., 2007].

The two algorithms called BIRCH and CURE are being used in large databases. This research paper emphasis on comparative study of BIRCH and CURE algorithms and a

data set has been taken to simulate both algorithms with help of Weka 3.6.9. Clustering output can be produced by many other tool like Octave, SPAETH2, XLMiner, DTREG, Orange, KNIME, Tanagra, Cluster3, CLUTO, KEEL, ELKI, but here we choose WEKA (Waikato Environment for Knowledge Analysis) for implementing our hierarchical clustering algorithm. Weka toolkit is a widely used toolkit for machine learning and data mining that was originally developed at the University of Waikato in New Zealand. It contains large collection of state-of-the-art machine learning and data mining algorithms written in Java. Weka has three principal advantages over most other data mining software. First, it is open source, which not only means that it can be obtained free, but more importantly it is maintainable, and modifiable, without depending on the commitment, health, or longevity of any particular institution or company. Second, it provides a wealth of state-of-the-art machine learning algorithms that can be deployed on any given problem. Third, it is fully implemented in Java and runs on almost any platform even a Personal Digital Assistant (PDA). There exist a comprehensive collection of data pre-processing and modelling techniques. WEKA has become very popular with the academic and industrial researchers, and is also widely used for teaching purposes [Abdullah H. Wahbeh et al., 2011].

II. BIRCH AND CURE HIERARCHICAL CLUSTERING ALGORITHM

2.1. BIRCH Algorithm

BIRCH (Balanced Iterative Reducing and Clustering Using Hierarchies) is an integrated agglomerative hierarchical clustering method. It is mainly designed for clustering large amount of metric data. It is mainly suitable when there is limited amount of main memory and have to achieve a linear I/O time requiring only in one database scan. It introduces two concepts, clustering feature and clustering feature tree (CF tree), which are used to summarize cluster representations [Tian Zhang et al., 1996].

A CF tree is a height-balanced tree that stores the clustering features for a hierarchical clustering. It is similar to B+-Tree or R-Tree. CF tree is balanced tree with a branching factor (maximum number of children per none leaf node) B and threshold T. Each internal node contains a CF triple for each of its children. Each leaf node also represents a cluster and contains a CF entry for each sub cluster in it. A sub cluster in a leaf node must have a diameter no greater than a given threshold value (maximum diameter of sub-clusters in the leaf node) [Tian Zhang et al., 1996].

An object is inserted to the closest leaf entry (sub cluster). A leaf node represents a cluster made up of all sub clusters represented by its entries. All the entries in a leaf node must satisfy the threshold requirements with respect to the threshold value T, that is, the diameter of the sub cluster must be less than T. If the diameter of the sub cluster stored in the leaf node after insertion is larger than the threshold value, then the leaf node and other nodes are split. After the insertion of the new object, information about it is passed toward the root of the tree. The size of the CF tree can be changed by modifying the threshold. These structures help the clustering method achieve good speed and scalability in large databases. Birch is also effective for incremental and dynamic clustering of incoming objects. This algorithm can find approximate solution to combinatorial problems with very large data sets [Harrington & Salibián-Barrera, 2010].

Figure 1 represents an overview of BIRCH algorithm. It mainly consists of four phases as shown in figure. The main task of phase 1 is to scan all data and build an initial memory CF tree using the given amount of memory and recycling space on disk. This CF tree tries to reflect the clustering information of the dataset as fine as possible under the memory limit. With crowded data points grouped as fine sub-clusters, and sparse data points removed as outliers, this phase creates a memory summary of the data. BIRCH first performs a pre-clustering phase in which dense regions of points are represented by compact summaries means by cluster tree, and then a hierarchical algorithm is used to cluster the set of summaries. BIRCH is appropriate for very large datasets, by making the time and memory constraints explicit.

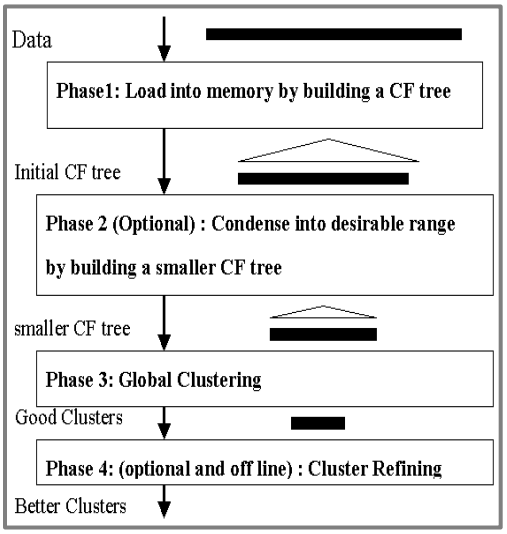


Figure 1: Overview of BIRCH Algorithm

2.2. CURE Clustering Algorithm

CURE (Clustering Using REpresentatives) is improved hierarchical clustering algorithm. Clustering, in data mining, is useful for discovering groups and identifying interesting distributions in the underlying data. Traditional clustering algorithms either favor clusters with spherical shapes and similar sizes, or are very weak in the presence of outliers. Here a clustering algorithm called CURE that is more robust to outliers, and identifies clusters having non-spherical shapes and wide variances in size. CURE achieves this by representing each cluster by a certain fixed number of points that are generated by selecting well scattered points from the cluster and then shrinking them toward the center of the cluster by a specified fraction. Having more than one

representative point per cluster allows CURE to adjust well to the geometry of non-spherical shapes and the shrinking helps to dampen the effects of outliers. To handle large databases, CURE employs a combination of random sampling and partitioning [Sudipto Guha et al., 1998].

CURE employs a novel hierarchical clustering algorithm that adopts a middle ground between centroid-based and representative-object based approaches. Instead of using a single centroid or object to represent a cluster, a fixed number of representative points in space are chosen. The representative points of a cluster are generated by first selecting well-scattered objects for the cluster and then "shrinking" or moving them toward the cluster center by a specified fraction, or shrinking factor. At each step of the algorithm, the two clusters with closest pair of representative point are chosen. Having more than one representative point per cluster allows CURE to adjust well to the geometry of non spherical shapes. The shrinking or condensing of clusters helps dampen the effects of outliers. Therefore, CURE is more robust to outliers and identifies clusters having non spherical shapes and wide variance in size. That's why it scales well for large databases without sacrificing clustering quality [Seema Maitrey et al., 2012].

A random sample drawn from the data set is first partitioned and each partition is partially clustered. The partial clusters are then clustered in a second pass to yield the desired clusters. It will be confirmed that the quality of clusters produced by CURE is much better than those found by other algorithms. Furthermore, it was demonstrated that random sampling and partitioning enable CURE to not only outperform existing algorithms but also to scale well for large databases without sacrificing clustering quality.

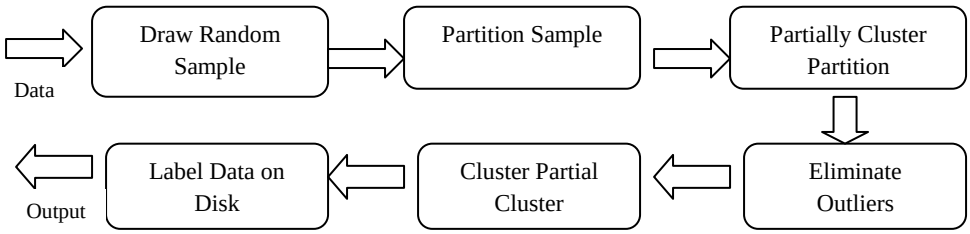
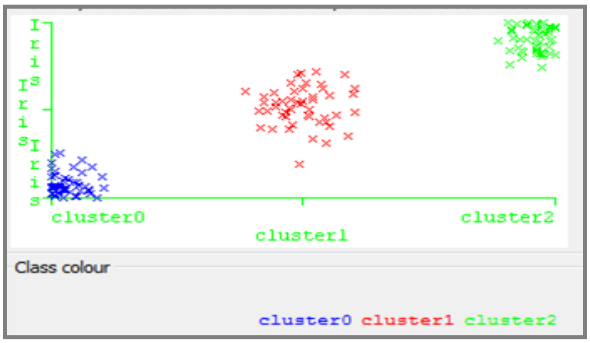


Figure 2: Overview of CURE Algorithm

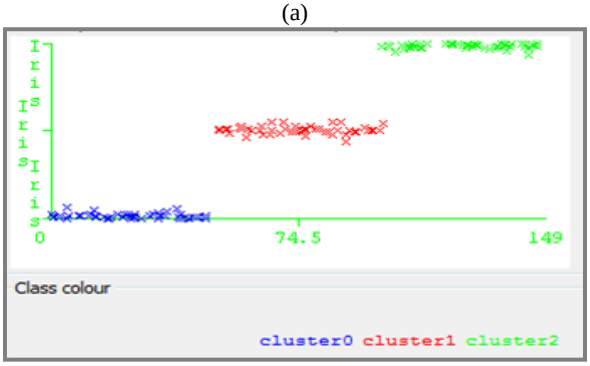
III. EXPERIMENT FOR BIRCH AND CURE IN WEKA 3.6.9

For our comparative study of BIRCH and CURE algorithm, Iris Plant Dataset is used. The Iris flower data set or Fisher's Iris data set is a multivariate data set introduced by Sir Ronald Fisher (1936). The data set contains 3 classes of 50 instances each, where each class refers to a type of iris plant. There are three flower types (classes): Setosa, Virginica, Versicolour and Four (non-class) attributes: Sepal width and length and Petal width and length. Here we take dataset in CSV (Comma Separated Value) file format.

BIRCH process begins by partitioning objects hierarchically using tree structure and then applies other clustering algorithm to refine the cluster. For implementing our algorithm, firstly we start Weka 3.6.9 data mining tool and load our data set (Iris plant data set.csv) file by pressing open file tab in Explorer panel and also we apply Unsupervised Add Cluster filtering on this data set. After that, we form Cluster Feature (CF) tree by pressing classifier tab and after tree formation we apply hierarchical clustering algorithm and then choose from cluster tab hierarchical clusterer and set number of cluster 3. In the end from cluster mode we choose classes to cluster evaluation and then press start button and we have cluster output and can see cluster formation of BIRCH algorithm in the form of graph by going to visualize panel. Firstly we have Cluster Tree formation. After that in next phase agglomerative hierarchical clustering algorithm applied on this cluster tree and we have result as shown below.



(a)



(b)

Figure 3: Cluster Formation of (a) BIRCH and (b) CURE

CURE is a hierarchical clustering algorithm that uses a fixed number of points as representatives. It presents a hybrid of two Partitioning + hierarchical clustering. Basically CURE is a hierarchical clustering algorithm that uses Partitioning. For implementing firstly we have to load our dataset same as performed in BIRCH process. After loading we perform our main step of partitioning by applying partitioning algorithm

such as k-means and the same number of cluster as previous algorithm which is 3. Next step is to done clustering of partial cluster that have formed in partitioning step. For this we apply hierarchical clustering and Mode of Cluster for performing clustering should be classes to cluster evaluation. On pressing start button, it will start formation of clusters on partitioned data and we have clusterer output. By right click on this we can visualize cluster assignment. Then we have result which shows the cluster visualization of CURE process in the form of graph as shown in above Figure3.

IV. COMPARATIVE STUDY OF BIRCH AND CURE ALGORITHMS

Both BIRCH and CURE process have its own advantages yet by comparison it is concluded that CURE is much better than BIRCH. An overview of the main characteristics is presented in a comparative way. Study is based on the following features of the algorithms as shown below:

- Complexity
- Input Parameters
- The type of the data that an algorithm supports (numerical, nominal)
- The shape of clusters
- Ability to handle noise and outliers
- Pre-clustering Phase
- The type of model that an algorithm presents
- Running Time
- Sensitivity towards Data input Order

Table 1 below summarizes the main concepts and the characteristics of the BIRCH and CURE Clustering algorithms**:**

Table 1: Comparative Table of BIRCH and CURE Clustering Algorithm

S. No.	Properties	BIRCH	CURE
1.	Complexity	O (n^2)	O (n^2)
2.	Input Parameter	Dataset is loaded in memory by building CF-tree	Random Sampling is done on the Dataset
3.	Large Dataset Handling	Yes	Yes
4.	Geometry	Shapes of cluster produced is almost spherical or convex of uniform size	Cluster produced can be of any arbitrary shape and in wide variance in sizes
5.	Noise	Handle noise effectively	Comparatively less sensitive towards noise handling
6.	Outlier Handling	Filtering of Outliers contained in dataset is done less effectively	Filtering of Outliers contained in dataset is done more effectively due to shrinking factor
7.	Type of Data Values	Numerical only	Numerical and Nominal both
8.	Type of Model	Dynamic or incremental model	Static Model
9.	Pre-clustering Phase	Cluster- Feature tree is formed	Partitioning is to be done on sample
10.	Running Time	Comparatively more	Partitioning improves running time by 50%
11.	Data Input Order Sensitivity	Yes	No

V. CONCLUSION

In this research paper, study is being done on the BIRCH and CURE clustering algorithms. Both BIRCH and CURE are hierarchical clustering approaches and both are used for large dataset. Comparison is made using Iris Plant dataset and Weka 3.6.9 data mining tool and the comparative results are given in the form of table. This comparative study is performed on the basis of parameters such as complexity, geometry, noise, running time etc. Both the algorithms have some advantages over the other yet by comparison it is concluded that quality of clusters produced by CURE is better than BIRCH. A combination of random sampling and partitioning allows CURE to handle large dataset more effectively. BIRCH identifies only convex or spherical cluster of uniform size while CURE can identifies clusters of non spherical shapes and wide variances in size using a set of representative's point for each cluster.

REFERENCES

[1] Tian Zhang, Raghu Ramakrishnan & Miron Linvy (1996), "BIRCH: An Efficient Data Clustering Method for Large Databases", *Proceedings of 1996 ACM-SIGMOD, International Conference on Management of Data*, Montreal, Quebec.

[2] Sudipto Guha, Rajeev Rastogi & Kyuseok Shim (1998), "CURE: An Efficient Clustering Algorithm for Large Databases", *Proceedings of 1998 ACM-SIGMOD International Conference on Management of Data*.

[3] I.K. Ravichandra Rao (2003), "Data Mining and Clustering Techniques", *DRTC Workshop on Semantic Web*, Bangalore.

[4] Jiawei Han & Micheline Kamber (2006), "Data Mining: Concepts and Techniques", *The Morgan Kaufmann/Elsevier India*.

[5] J.A.S. Almeida, L.M.S. Barbosa, A.A.C.C. Pais & S.J. Formosinho (2007), "Improving Hierarchical Cluster Analysis: A New Method with Outlier Detection and Automatic Clustering", *Chemometrics and Intelligent Laboratory Systems*, Vol. 87, Pp. 208–217.

[6] J. Harrington & M. Salibián-Barrera (2010), "Finding Approximate Solutions Combinatorial Problems with Very Large Data Sets using BIRCH", *Computational Statistics and Data Analysis*, Vol. 54, Pp. 655–667.

[7] Abdullah H. Wahbeh, Qasem A. Al-Radaideh, Mohammed N. Al-Kabi & Emad M. Al-Shawakfa (2011), "A Comparison Study between Data Mining Tools over some Classification Methods", *International Journal of Advanced Computer Science and Applications (IJACSA), Special Issue on Artificial Intelligence*.

[8] Seema Maitrey, C.K. Jha, Rajat Gupta & Jaiveer Singh (2012), "Enhancement of CURE Clustering Technique in Data Mining", *Proceedings in International Journal of Computer Application*, Published by Foundation of Computer Science, New-York, USA. April 2012.

Yogita Rani. Yogita Rani passed her M.Tech in Computer Science & Engineering in June 2013 from Department of Computer Science and Applications, Chaudhary Devi Lal University, Sirsa. She passed her B.Tech. in Computer Science & Engineering in June 2011 from Chaudhary Devi Lal Memorial Government Engineering College, Panniwala Mota, Sirsa.

Manju. Manju is Lecturer at Department of Computer Engineering, CDL Govt. Polytechnic Educaiton Society, Nathusari Chopta, Sirsa- 125055, Haryana (India). Presently she is pursuing her research as scholar of Doctorate of Phillosophy (CS) at Department of Computer Science and Applications, Kurukshetra University, Kurukshetra- 136119, Haryana (India). She received her Master of Computer Applications in 2005 from Kurukshetra University, Kurukshetra, her Master of Phillosophy (CS) in 2007 from Chaudhary Devi Lal University, Sirsa and her M. Tech (IT) from Karnataka State Open University, Mysore (India). Her area of interest includes Data Mining, Ant Colony Optimization and Swarm Intelligence.

Harish Rohil. Harish Rohil received his Ph.D. (CS) in the field of Repository Based Software Reuse, from Department of Computer Science and Applications, Kurukshetra University, Kurukshetra-136119, Haryana (India) in December 2012. He received his M.Tech (CSE) from Guru Jambheshwar University, Hisar-125001, Haryana (India) in 2004 and his Master of Science (Computer Sc./ Software) from Kurukshetra University, Kurukshetra in 2002. He is working as Assistant Professor at Department of Computer Science and Applications, Chaudhary Devi Lal University, Sirsa- 125055, Haryana (India) since July 2004. His area of interest includes Software Reuse, Data Structure, Data Mining, and Swarm Intelligence.