Journal Homepage: -www.journalijar.com

INTERNATIONAL JOURNAL OF ADVANCED RESEARCH (IJAR)

Article DOI:10.21474/IJAR01/11410
DOI URL: http://dx.doi.org/10.21474/IJAR01/11410

RESEARCH ARTICLE

CLUSTER ANALYSIS IN VIRTUALLY INTEGRATED ENVIRONMENTS

Lakshmi N

Manuscript Info

Manuscript History
Received: 28 May 2020
Final Accepted: 30 June 2020
Published: July 2020

Key words:-
BIRCH Clustering Algorithm, Virtually Integrated Computing

Abstract

Today data clustering has been widely applied to many practical applications like social network analysis, scientific analysis. Since there is an enormous amount of data generated by the Internet users and because of limited memory, there is a need for the design of clustering algorithms. In this paper, we perform cluster analysis in virtually integrated environments. In cluster analysis, we apply BIRCH (Balanced Iterative Reducing and Clustering using Hierarchies) algorithm on the dataset and perform global clustering and cluster refining to produce better clusters. This paper presents a way to implement BIRCH algorithm that is suitable for very large datasets and increase its efficiency by executing task in parallel. As a result, there is a linear growth of execution time with increase in dataset. The Information Technology (IT) industry is one among the most information and demanding industries. In the IT field, the finance department is the one where the knowledge and data keep on developing on a daily basis. It has been assessed that several companies are most affected by the inaccurate results produced by the softwares for calculating the financial data. Especially, the multi- national companies are mostly suffered due to the incorrect financial data available to them. The company has various branches all over the world. Each branch of the company has run different environments. One branch may have the network in the form of distributed environment and another branch may have the network in the form of cloud environment. Hence the data are heterogeneous in these different environments.

Introduction:-

Data clustering techniques are used in many applications, such as mobile data analysis and user grouping for marketing. The usage of integrated computing increases day by day in the area of data mining. The contemporary clustering algorithms are inefficient on the integrated computing environment. The time for execution of these clustering algorithms is very large and they are not suitable for highly distributed database. The integrated environment provides all of its resources as services and makes use of the well-established standards. The main enabling technology for integrated computing is virtualization. The virtualization software separates a physical computing service into one or more “virtual” devices, each of which can be easily used and managed to perform computing tasks. Virtualization provides the agility required to speed up IT operations, and reduces cost by increasing infrastructure utilization.

Corresponding Author:- Lakshmi N

The Information Technology (IT) industry is one among the most information and demanding industries. In the IT field, the finance department is the one where the knowledge and data keep on developing on a daily basis. It has been assessed that several companies are most affected by the inaccurate results produced by the softwares for calculating the financial data. Especially, the multi- national companies are mostly suffered due to the incorrect financial data available to them. The company has various branches all over the world. Each branch of the company has run different environments. One branch may have the network in the form of distributed environment and another branch may have the network in the form of cloud environment. Hence the data are heterogeneous in these different environments.

Data mining is a process of discovering meaningful patterns and relationships that are hidden in large dataset[1]. Data mining refers to extracting or "mining" knowledge from large amount of data[2]. Clustering is the process of grouping a set of objects into classes of similar objects. A cluster is a collection of data objects that are similar to one another within the same cluster and are dissimilar to the objects in other clusters. The contemporary clustering algorithms can handle large datasets but huge memory usage is always a concern. Hence, using integrated computing with virtualization concept is used to take care of more memory requirement very easily.

A hierarchical clustering method works by grouping data objects into tree of clusters. The hierarchical clustering method, though simple, often encounters difficulties regarding selection of merge or split points. Such a decision is critical because once a group of objects is merged or split, the process at the next step will operate on the newly generated clusters. It will neither undo what was done previously nor perform object swapping between clusters. Thus merge or split decisions, if not well chosen at some step, may lead to low-quality clusters. Moreover, the method does not scale well, because each decision to merge or split requires the examination and evaluation of a good number of objects or clusters.

One promising direction for improving the clustering quality of hierarchical methods is to integrate hierarchical clustering with other clustering techniques, resulting in multiple-phase clustering. That method is called BIRCH (Balanced Iterative Reducing and Clustering using Hierarchies), begins by partitioning objects hierarchically using tree structures, where the leaf or low-level non-leaf nodes can be viewed as "micro clusters" depending on the scale of resolution. It then applies other clustering algorithms to perform macro clustering on the micro clusters.

BIRCH overcomes the two difficulties of agglomerative clustering methods:

1. Scalability and
2. The inability to undo what was done in the previous step

BIRCH introduces two concepts, clustering feature and clustering feature tree (CF Tree), which are used to summarize cluster representations. These structures help the clustering method achieve good speed and scalability in large databases and also make it effective for incremental and dynamic clustering of incoming objects. To implement this algorithm, we require the integrated computing virtualized environment. Assuming, data is distributed among different nodes. Using virtualization concept, we create instances of each distributed node. This algorithm loads data from the nodes into memory by building a CF tree. We conclude that this algorithm performs global clustering and cluster refining to produce better clusters.

This paper is organized as follows. We briefly review the related works in section II. Section III describes the design of algorithm using BIRCH technique that suits for the integrated environment. Section IV deals with the results obtained in the proposed approach. Section V concludes the paper.

Related Works:

Chun - Chieh Chen et al., (Chun - Chieh Chen et al., 2015) proposed a distributed density – based clustering method for clustering large datasets in the heterogeneous cloud[3]. They devised an highly scalable algorithm called HiClus. Based on their experimental findings, applying GPU (Graphics processing Unit) to the clustering phase obtains better performance. When a dataset is larger, the runtime increases significantly. The algorithm running time is O(n*n) and O(n logn) for the datasets ranging from 50 K to 400 K. They run HiClus with maximum GPU thread support to effectively split data into balanced partitions. The overlapping partition is better for enlarging the boundary and it decreases the difficulties during the merging phase.

Ashish et al., (Ashish et al., 2014) proposed a clustering technique for data analysis across distributed environment[4]. They devised an alternative method for MapReduce, known as Bulk Synchronous Parallelism method. The method solves the problem of inefficiency when there are large number of datasets. The author used HAMA programming model to solve the in-efficient problem of clustering on large datasets. The execution time varies based on the total number of documents clustered. The proposed method using Bulk Synchronous Parallelism is quite efficient than the MapReduce method.

Sundararajan et al., (Sundararajan et al., 2014) proposed an approach for enhancing the data quality and for fixing the optimal number of clusters[5]. Here, k clusters are taken as input. They proposed a Modified Firefly algorithm to determine the centroid of the user specified number of clusters. The algorithm was further extended using dynamic k-means clustering to enhance centroids and clusters. The algorithm computing time is less when comparing with the k-means algorithm. They conclude that the proposed method finds the maximum number of clusters in less time with better cluster quality and increased optimality.

Luo Zhong et al., (Luo Zhong et al., 2014) proposed a study based on the mass data of the tunnel[6]. The author proposed an improved parallel clustering algorithm based on k-means clustering algorithm that uses the MapReduce within cloud computing that deals with the data. The author introduced the Good Center Parallel K-Means (GCPK) Algorithm which find out a point as the center, and can separate data on average basically. The computation time is based on the number of datasets. The author used MATLAB 2010 single-machine environment. This algorithm has a better clustering result than the traditional clustering algorithm.

Shraddha Masih et al., (Shraddha Masih et al., 2014) proposed a cloud based data mining model which provides the facility of mass data storage along with distributed data mining facility[7]. They proposed a generalized framework supporting Distributed Data Mining and Storage as a service on private networks. The required datasets can be selected and can be run with k-means algorithm which runs in a distributed fashion through MapReduce. The interface is designed using Java and Apache Hadoop is used for creating multimode setup. The main idea of this study is to define k-centroids, one for each cluster. When the datasize is large, Owncloud service is used for storage.

Kriti Srivastava et al., (Kriti Srivastava et al., 2013) proposed a way to implement Hierarchical Agglomerative Clustering (HAC) Algorithm that makes suitable for very large datasets[8]. They presented a Modified Hierarchical Agglomerative Clustering algorithm for processing datasets. Firstly, the virtual k-means is applied at layer 1. Secondly, the merging of files will be done and thirdly, the modified hierarchical agglomerative algorithm is used. The above concept is implemented in cloud architecture that requires master and slave nodes. The nodes have MySQL and Java installed to it. The datasets are stored in MySQL and the modified algorithm is written in Java. The algorithm efficiency is based on the total number of data mined as nodes increase, the data also increases. They conclude that this algorithm handles large dataset, and increases the efficiency.

Design Of Algorithm Using Birch Technique:

BIRCH (Balanced Iterative Reducing and Clustering using Hierarchies), is an efficient centroid-based clustering method that aims to reduce memory-related overheads of the clustering process. Using contemporary clustering algorithms, more memory is needed. The clusters in BIRCH are built incrementally by scanning the dataset and inserting each data point to the closest cluster. These insertions amount to simple updates of the CF (Clustering Feature) of the cluster. However, the clusters are constrained to have a bounded diameter, and if no cluster can absorb a data point, the scan creates a new singleton cluster.

Clustering Feature (CF):

BIRCH attempts to minimize the memory requirements of large datasets by summarizing the information contained in dense regions as Clustering Feature (CF) entries.

CF Definition : *Given N d-dimensional data points in a cluster: $\{\vec{X}_i\}$ where $i = 1, 2, ..., N$, the* **Clustering Feature (CF)** *entry of the cluster is defined as a triple:* $\mathbf{CF} = (N, \vec{LS}, SS)$, *where N is the number of data points in the cluster, $\vec{LS}$ is the linear sum of the N data points, i.e., $\sum_{i=1}^{N} \vec{X}_i$, and SS is the square sum of the N data points, i.e., $\sum_{i=1}^{N} \vec{X}_i^{\,2}$.*

Here, the subcluster is equal to the sum of the CFs.

CF Additivity Theorem : *Assume that* $\mathbf{CF_1} = (N_1, \vec{LS}_1, SS_1)$, *and* $\mathbf{CF_2} = (N_2, \vec{LS}_2, SS_2)$ *are the CF entries of two disjoint subclusters. Then the CF entry of the subcluster that is formed by merging the two disjoint subclusters is:*

$$\mathbf{CF_1} + \mathbf{CF_2} = (N_1 + N_2, \vec{LS}_1 + \vec{LS}_2, SS_1 + SS_2) \quad (11)$$

CF-tree:

The CF-tree is a very compact representation of the dataset because each entry in a leaf node is not a single data point but a sub-cluster. Each non-leaf node contains at most B entries. In this context, a single entry contains a pointer to a child node and a CF made up of the sum of the CFs in the child (sub-clusters of sub-clusters). On the other hand, a leaf node contains at most L entries, and each entry is a CF (sub-clusters of data points). All entries in a leaf node must satisfy a threshold requirement. That is to say, the diameter of each leaf entry has to be less than Threshold. In addition, every leaf node has two pointers, prev and next, which are used to chain all leaf nodes together for efficient scans.

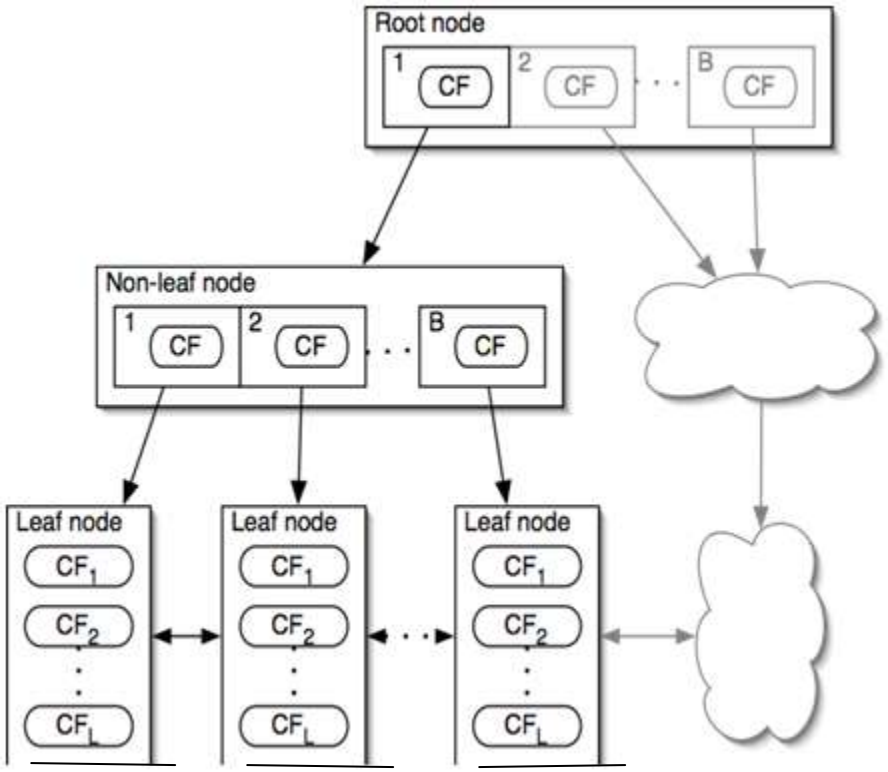

Figure 3.1:- A CF tree structure for the BIRCH technique.

Insertion Algorithm:

The CF entry (a single data point or sub-cluster) can be inserted into a CF-tree. The following steps has to be taken to insert a CF entry into the CF tree.

Identify the appropriate leaf:

Starting from the root, recursively descend the DF-tree by choosing the closest child node according to the chosen distance metric.

Modify the leaf:

Upon reaching a leaf node, find the closest entry and test whether it can absorb the CF entry without violating the threshold condition. If it can, update the CF entry, otherwise, add a new CF entry to the leaf. If there isn't enough space on the leaf for this new entry to fit in, then we must split the leaf node. Node splitting is done by choosing the two entries that are farthest apart as seeds and redistributing the remaining entries based on distance.

Modify the path to the leaf:
Recall how every non-leaf node is itself a CF composed of the CFs of all its children. Therefore, after inserting a CF entry into a leaf, we update the CF information for each non-leaf entry on the path to the leaf. In the event of a split, we must insert a new non-leaf entry into the parent node and have it point to the newly formed leaf. If according to B, the parent doesn't have enough room, then we must split the parent as well, and so on up to the root.

Clustering Algorithm:
The various phases in the clustering algorithm will be as follows.

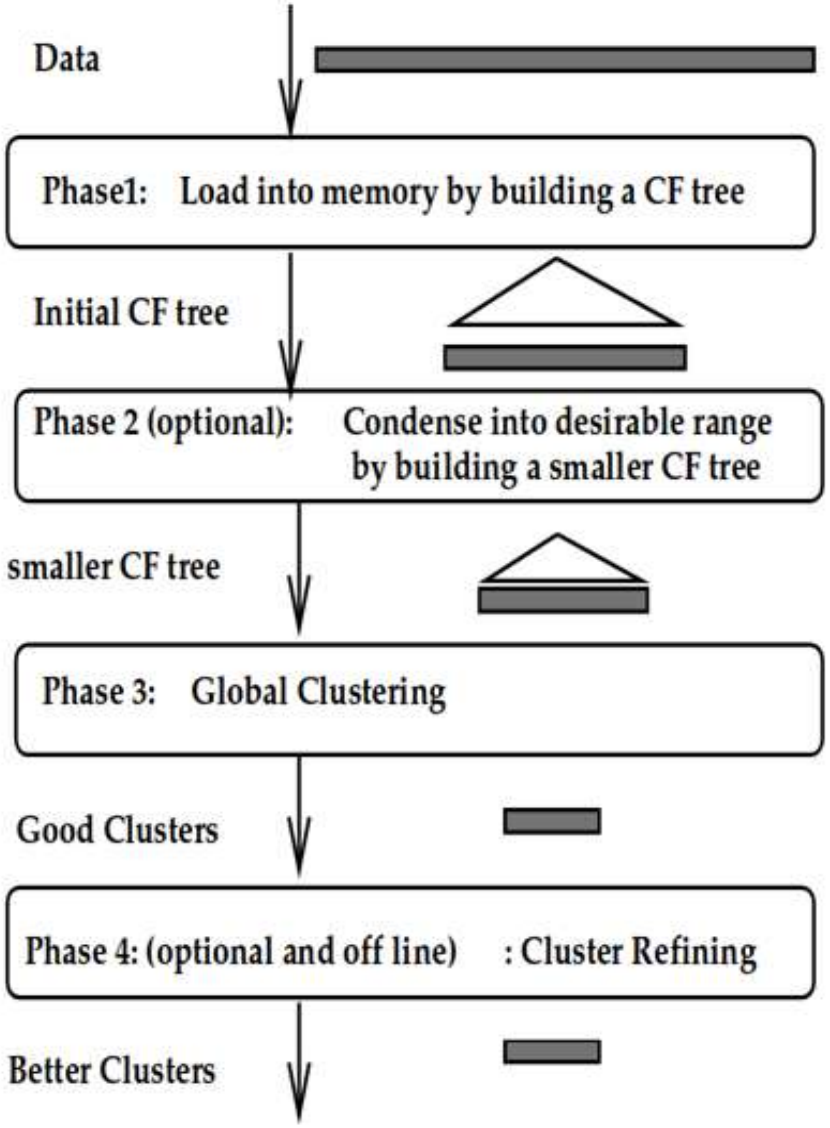

Figure 3.2:- Different phases of producing better clusters.

Phase 1:
The algorithm starts with an initial threshold value, scans the data, and inserts points into the tree. If it runs out of memory before it finishes scanning the data, it increases the threshold value, and rebuilds a new, smaller CF-tree, by re-inserting the leaf entries of the old CF-tree into the new CF-tree. After all the old leaf entries have been re-inserted, the scanning of the data and insertion into the new CF-tree is resumed from the point at which it was interrupted.
A good choice of threshold value can greatly reduce the number of rebuilds. However, if the initial threshold is too high, we will obtain a less detailed CF-tree than is feasible with the available memory.

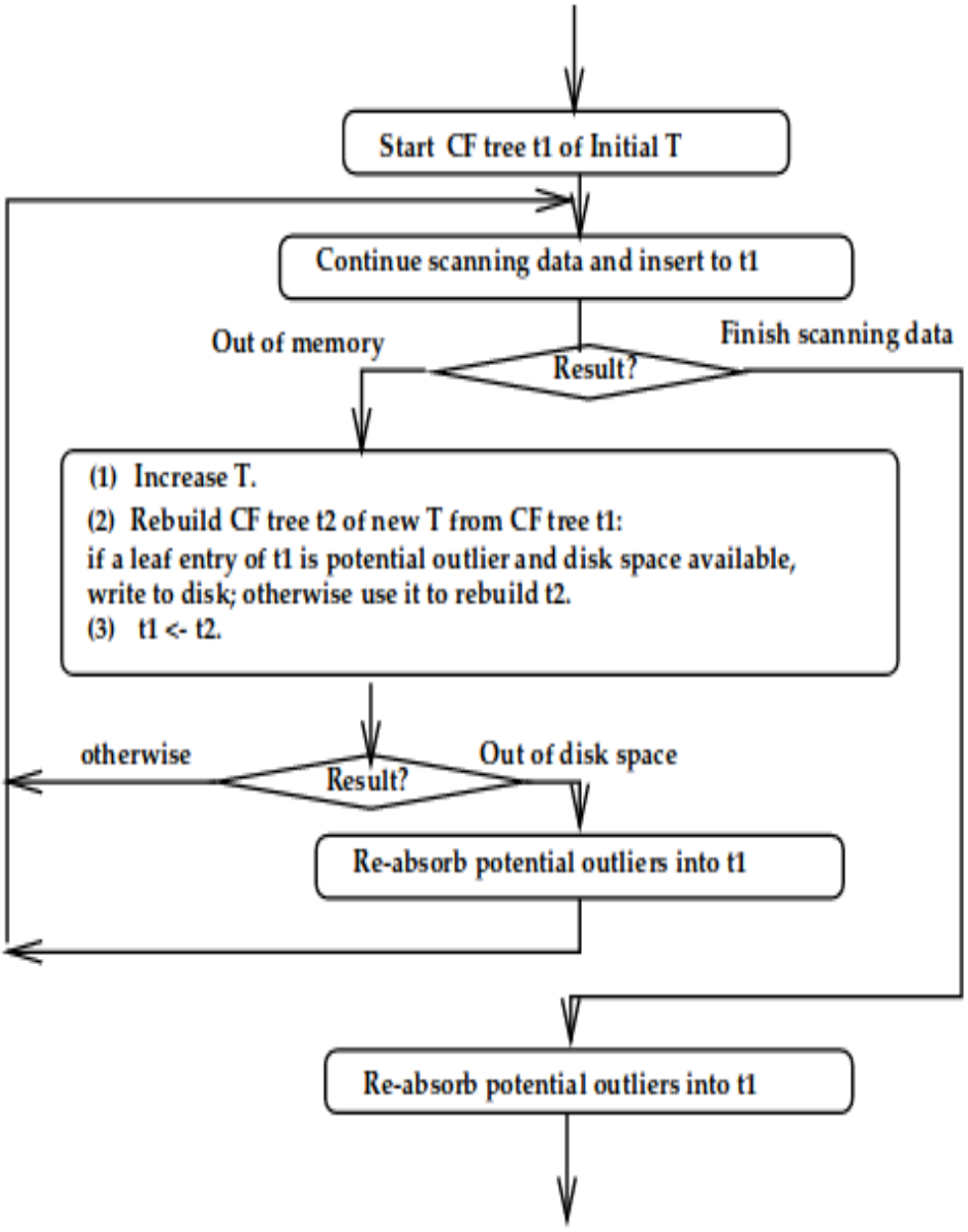

Figure 3.3:- Flowchart showing steps in building a CF Tree.

Some fixed amount of disk space is allocated for handling outliers. Outliers are leaf entries of low density that are judged to be unimportant with respect to the overall clustering pattern. While rebuilding the CF-tree by reinserting the old leaf entries, the size of the new CF-tree is reduced in two ways. First, allowing to increase the threshold value, thereby allowing each leaf entry to absorb more points. Second, some leaf entries are treated as potential outliers and write them out to disk. An old leaf entry is considered a potential outlier if it has far fewer data points than average. An increase in the threshold value or a change in the distribution in response to the new data could well mean that the potential outlier no longer qualifies as an outlier. In consequence, the potential outliers are scanned to check if they can be re-absorbed in the tree without causing the tree to grow in size.

Phase 2:

Given that certain clustering algorithms perform best when the number of objects is within a certain range, the crowded sub-clusters are grouped into larger ones resulting in an overall smaller CF-tree.

Phase 3:
Almost any clustering algorithm can be adapted to categorize Clustering Features instead of data points. For instance, KMEANS is used to categorize our data, all the while deriving the benefits from BIRCH (i.e. minimize I/O operations).

Phase 4:
Although, the tree may have been rebuilt multiple times, the original data has only been scanned once. Phase 4 involves additional passes over the data to correct inaccuracies caused by the fact that the clustering algorithm is applied to a coarse summary of the data. Phase 4 also provides us with the option of discarding outliers.

Summary:
BIRCH clustering algorithm with a new dissimilarity measure is used for warehouse large heterogeneous databases. Using a dissimilarity measure, each object with the modes was compared and each object was allocated to the adjacent cluster. After the distribution of each object to the clusters, the mode of the cluster was updated. Thus all the similar objects were placed in one cluster. Then the classification was done with the help of fuzzy logic. Later, the user can simply gather the appropriate IT financial data to offer the essential information in a direct, speedy and significant way. This technique assures that the IT data warehouse is a beneficial technique for supporting financial data analysis. This approach will be one of the imperative data sources for IT data mining. The technique increase the speed of query processing and reduced the mining cost.

Experimental Setup And Result Analysis:
The implementation of the study requires master and slave nodes since the study is based on the cluster analysis in the integrated environment. The master -slave architecture can be formed by using Python in the master and the slave nodes. The data used in this experiment is taken from the DATASET IT Sector Financial Data. This study uses Python for plotting the values.

The sample data from the dataset IT sector Financial data can be shown as follows:

Training costs	Bandwidth Costs	Infrastructure Investment	IT consumables	Is Multinational Company	Software licensing costs	IT Staff remuneration	IT Financial Policy
2	50	12500	98	1	50	98	98
0	13	3250	28	1	13	28	28
1	16	4000	35	1	16	35	35
2	20	5000	45	1	20	45	45
1	24	6000	77	0	24	77	77
4	4	1000	4	0	4	4	4
2	7	1750	14	1	7	14	14
1	12	3000	35	0	12	35	35
2	9	2250	22	1	9	22	22
5	46	11500	98	1	46	98	98
4	23	5750	58	0	23	58	58
0	3	750	4	0	3	4	4
2	10	2500	28	1	10	28	28
1	13	3250	47	0	13	47	47
2	6	1500	15	1	6	15	15
2	5	1250	11	1	5	11	11
2	14	3500	48	1	14	48	48

Figure 4.1:- Sample data from the dataset IT sector Financial data.

Python Tool:

Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. Its high-level built in data structures, combined with dynamic typing and dynamic binding, make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together. Python's simple, easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance. Python supports modules and packages, which encourages program modularity and code reuse. The Python interpreter and the extensive standard library are available in source or binary form without charge for all major platforms, and can be freely distributed.

The Pandas module of Python is used to clean and restructure the data. Pandas is an open-source module for working with data structures and analysis, one that is ubiquitous for data scientists who use Python. It allows for data scientists to upload data in any format, and provides a simple platform organize, sort, and manipulate that data.

In Machine Learning, the types of Learning can broadly be classified into three types: 1. Supervised Learning, 2. Unsupervised Learning and 3. Semi-supervised Learning. Algorithms belonging to the family of Unsupervised Learning have no variable to predict tied to the data. Instead of having an output, the data only has an input which would be multiple variables that describe the data. This is where clustering comes in.

Result Analysis:-

The BIRCH algorithm has been implemented using python language and the pandas is used as the data science language. Some sample coding implementing BIRCH in Python is shown as follows.

```
import numpy as np from matplotlib

importpyplot as plt

import seaborn as sns

sns.set() from sklearn.datasets.samples_generator
import make_blobs from sklearn.cluster
 import Birch
```

Scikit-learn is used to generate data with nicely defined clusters.

```
X, clusters = make_blobs(n_samples=450, centers=6, cluster_std=0.70, random_state=0)
plt.scatter(X[:,0], X[:,1], alpha=0.7, edgecolors='b')
```

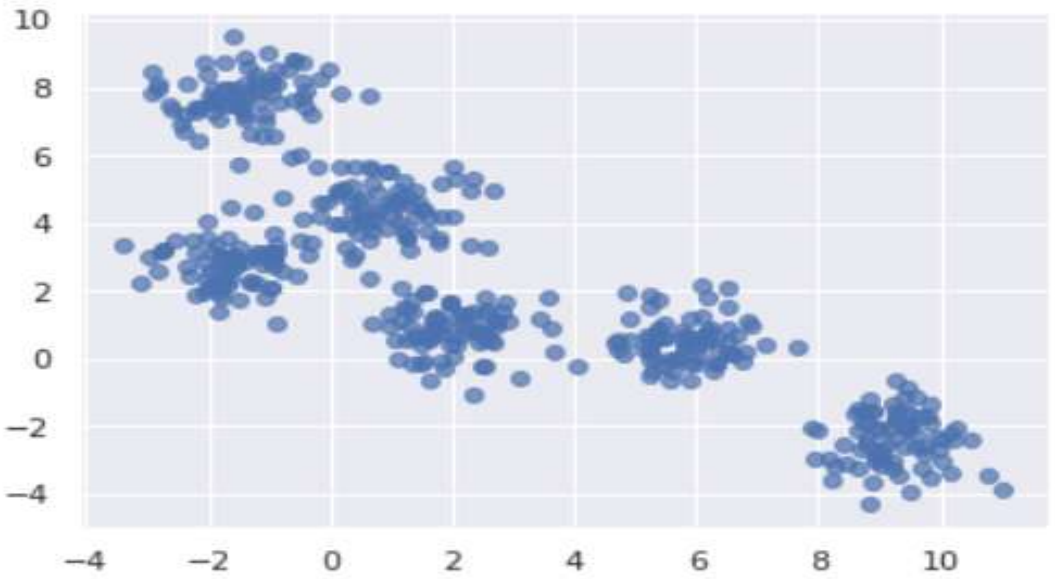

Figure: 4.2:- Generating data with clusters using scikit-learn.

In the next step, initializing and training the model, using the following parameters:
Threshold:
The radius of the sub-cluster obtained by merging a new sample and the closest sub-cluster should be lesser than the threshold.

Branching_factor:
Maximum number of CF sub-clusters in each node

N_clusters:
Number of clusters after the final clustering step, which treats the sub-clusters from the leaves as new samples. If set to 'None', the final clustering step is not performed and the sub-clusters are returned.

brc = Birch(branching_factor=50, n_clusters=None, threshold=1.5)brc.fit(X)
The 'predict' method is used to obtain a list of points and their respective cluster.

labels = brc.predict(X)

Finally, the data points are plotted using a different color for each cluster.

plt.scatter(X[:,0], X[:,1], c=labels, cmap='rainbow', alpha=0.7, edgecolors='b')

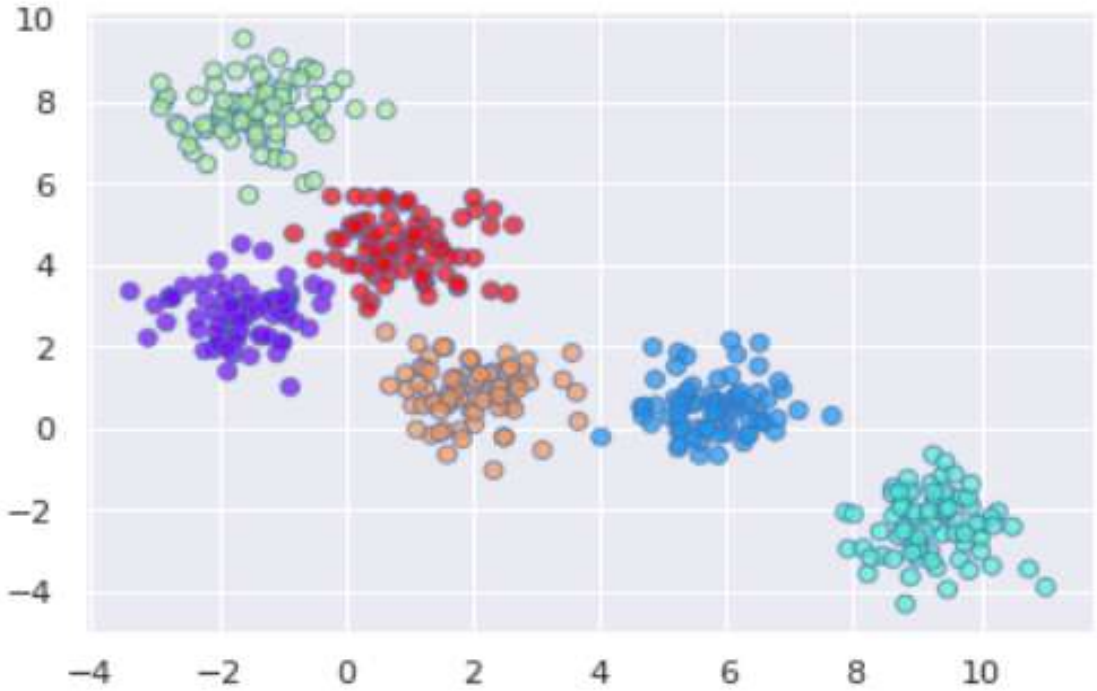

Figure: 4.3:- Plotting datapoints using different colours.

Summary:
BIRCH provides a clustering method for very large datasets. It makes a large clustering problem plausible by concentrating on densely occupied regions, and creating a compact summary. BIRCH can work with any given amount of memory, and the I/O complexity is a little more than one scan of data. Other clustering algorithms can be applied to the sub-clusters produced by BIRCH.

Conclusion:-

Clustering is the most important unsupervised classification technique. The Hierarchical agglomerative clustering method is not suited for clustering of large datasets. Hence BIRCH technique is used here for clustering of very large financial data in IT sector from the various environments. The proposed approach finds the optimum number of clusters effectively during execution using BIRCH technique. Hence the cluster quality is improved. This thesis

gives the overall financial costs for the categorized financial parameters like software licensing costs, training costs, telecommunication costs, system implementation costs, IT consumables, IT staff remuneration, etc of the particular organization. Here, the BIRCH technique is used to overcome the scalability problem which was faced with the hierarchical method. Hence this thesis helps to cluster the data from the large integrated environments.

For further enhancements, we planned to use some advanced clustering techniques for clustering large datasets in integrated environment. Some advanced clustering techniques would be used to cluster the financial data of the IT sector from different kinds of environments globally.

References:-

1. U. Fayyad, G. P. Shapiro, and P. Symth, " From data mining to knowledge discovery in databases," A Magazine 37-53
2. Data Mining and Analytics Resources. Available: http://www.kdnuggets.com
3. Chun-Chieh Chen, and Ming-Syan Chen, " HiClus: Highly Scalable Density-based Clustering with Heterogeneous Cloud," INNS Conference on Big Data, 2015
4. Ashish A. Golghate, Shailendra W. Shende, " Parallel K-Means Clusteing Based on Hadoop and Hama," International Journal of Computing and Technology, 2014
5. Sundararajan S. and Karthikeyan S., " An Efficient Hybrid Approach for Data Clustering Using Dynamic K-Means Algorithm and Firefly Algorithm," ARPN Journal of Engineering and Applied Sciences, 2014
6. Luo Zhong, KunHao Tang, Lin Li, Guang Yang, and JingJing Ye, " An Improved Clustering Algorithm of Tunnel Monitoring Data for Cloud Computing," The Scientific World Journal, 2014
7. Shraddha Masih, Sanjay Tanwani, " Distributed Framework for Data Mining As a Service on Private Cloud," Int. Journal of Engineering Research and Applications, 2014
8. Kriti Srivastava, R. Shah, D. Valia and H. Swaminarayan, " Data Mining Using Hierarchical Agglomerative Clustering Algorithm in Distributed Cloud Computing Environment," International Journal of Computer Theory and Engineering, 2013
9. https://www.kdnuggets.com/datasets "Datasets for Data Mining and Data Science".