Enhancing the Efficiency and Scalability of Big Data Using Clustering Algorithms
Kosha Kothari\textsuperscript{1} Ompriya Kale\textsuperscript{2}
\textsuperscript{1,2}M.E Student
\textsuperscript{1,2}LJ Institute of Technology, Ahmedabad, Gujarat, India

Abstract— The data that has been produced by numerous scientific applications and incorporated environment has grown rapidly not only in size but also in variety in current era. The data collected is of very large amount and there is an adversity in gathering and evaluating such big data. The main goal of clustering is to categorize data into clusters such that objects are grouped in the same cluster when they are “similar” according to similarities, traits and behavior. The effectiveness and efficiency of the existing algorithms is, somewhat limited, since clustering with big data requires clustering high-dimensional feature vectors and since big data often contain large amounts of noise and large datasets. K-Means and DBSCAN are complement to analyze big data on cloud environment. A hybrid approach based on parallel K-Means and parallel DBSCAN is proposed to overcome the drawbacks of both these algorithms. The hybrid approach combines the benefits of both the clustering techniques. The proposed technique is evaluated on the MapReduce framework of Hadoop Platform. The results show that the proposed hybrid approach is an improved version of parallel K-Means clustering algorithm and parallel DBSCAN algorithm.

Key words: Clustering Algorithms, Unsupervised learning, Data Mining, K-Means, DBSCAN, Big data

I. INTRODUCTION

The term big data is considered as an umbrella which includes everything from digital data to health data. Big data has evolved from various stages starting from primitive and structured data to complex relational data and now very complex and unstructured data. The concept of how data became big started seventy years ago when the growth rate in volume of data was known as information explosion. In 1944 Fermont Rider, a librarian estimated that size of American Universities libraries is getting doubled every sixteen years. He estimated that by this growth rate there would be 200,000,000 volumes by 2040. In decade of 90 IBM introduced the relational database concept in which data can be stored in tables and can be analyzed easily by using different analysis techniques. By the end of 2003 there was 5 exabyte of data that was created; today this amount of information is created in just two days. This data is generated from different sources and includes online transactions, sensor data, social networking, health records, census and science data and live streaming data. This digital data is of 2.72 zettabytes and is predicted to be doubled every two years [1].

II. BIG DATA AND DATA ANALYTICS

Nowadays the data collected from different sources is not only growing in its size but is also increasing in variety and variability. The big data is defined as datasets whose size is beyond the ability of typical database tools to store, capture, manage and analyze. The best way to define big data is via three Vs which are data volume, data velocity and data variety or variability [2]. The data volume is regarding size (terabytes and petabytes), records, transactions and tables, files. The data velocity is about how frequently data is generated by an application in real time or in streaming. The data variety includes different types of data that is structured (simply RDBMS), semi-structured (XML, RSS feeds) and unstructured (text, human languages). Big data is remarkably diverse in terms data types, sources and entities represented. Data analytics has evolved lot over the years and Big Data Analytics is the latest in this evolution of data. Big Data Analytics is the process of analyzing large amount of data having different variety to uncover unknown correlation, hidden patterns and other useful information. Such information results in business benefits, such as increased revenue and effective marketing.

The primary goal of big data analytics is to help enterprises make better business decisions and other users to analyze huge volumes of transaction data as well as other data which is left by Business Intelligence (BI) programs. Big data analytics can be done with advanced analytics software tools such as data mining and predictive analytics. But big data collected from unstructured data sources is not fit in the traditional data warehouses which lead to new big data technology. These technology associated with big data analytics includes Hadoop, MapReduce and NoSQL databases. Large datasets across the clustered systems can be processed from these technologies. The pitfalls for organizations on big data analytics includes high cost for hiring professionals and challenges in integrating Hadoop and data warehouses [3].

III. CRITERION TO BENCHMARK CLUSTERING METHODS

Distinct criteria need to be used to figure out the relative strengths and weaknesses of every algorithm when surveying clustering methods for big data, with respect to the three-dimensional properties of big data, including Volume, Velocity, and Variety. There are certain characteristics of big data which are listed below [4]:

(1) Volume – The volume is related to the size of data. At present data is in petabytes and in near future it will be of zettabytes. It is the most distinct feature that demands specific requirements to all classical technologies and tools used. To guide the selection of a suitable clustering algorithm with respect to the Volume property, the following criteria are considered: (i) size of the dataset, (ii) handling high dimensionality and (iii) handling outliers/noisy data.

(2) Variety – It refers to the ability of a clustering algorithm to handle different types of data
(numerical, categorical and hierarchical). It deals with the complexity of Big data [5]. To guide the selection of a suitable clustering algorithm with respect to the Variety property, the following criteria are considered: (i) Type of dataset and (ii) clusters shape. The data is not coming from single source it includes semi structured data like web pages, log files etc, raw, and structured and unstructured data.

3) Velocity – refers to the speed of a clustering algorithm on big data. Big Data are generated at high speed [5]. To guide the selection of a suitable clustering algorithm with respect to the Velocity property [6], the following criteria are considered: (i) Complexity of algorithm and (ii) the run time performance.

4) Variability – The variability considers inconsistencies in data flow.

5) Value – The value is importance of data used by the user. The user queries against certain data stored, obtains result, rank them and can store for future work.

6) Complexity – The data is coming from various resources in huge amount thus it is difficult to link or correlate multiple data.

The correspondent criterion of big data is explained in detail:

- Type of dataset: Most of the classical clustering algorithms focus either on numeric data or on categorical data. But, the data that is been collected in the real world contains both numeric and categorical attributes. It is difficult for applying classical clustering algorithm directly into these kinds of data. Clustering algorithms work effectively either on purely numeric data or on purely categorical data; most of them perform poorly on mixed categorical and numerical data types.

- Size of dataset: The size of the dataset has a dominant effect on the quality of clustering. When the data size is small some clustering methods are more efficient clustering methods than others.

- Handling outliers/ noisy data: An effectual algorithm will often be able to handle outlier/noisy data because the data in most of the real applications are not pure. Also, noise will make it difficult for any algorithm to cluster an object into a suitable cluster. Thus it will affect the results that are provided by the algorithm.

- Stability: One of the important features for any clustering algorithm is the ability to generate the same partition of the data irrespective of the order in which the patterns are presented to the algorithm. Thus the stability of the algorithm should be quite effective.

- Cluster shape: A good clustering algorithm should be able to handle clusters of arbitrary shape that are produced by real data and their wide variety of data types [6].

- Input parameter: An alluring feature for clustering is the one that has fewer parameters, since a large number of parameters may affect cluster quality because they will depend on the values of the parameters.

![Fig. 3.1: Characteristics of Big Data [7]](https://example.com/fig3.1.png)

- Handling high dimensionality: This is particularly important feature in cluster analysis because many applications require the analysis of objects containing a large number of features. For example, any text documents might contain thousands of terms or keywords as features. Thus it becomes complex due to the curse of dimensionality. Many dimensions may not be relevant. The data becomes increasingly sparse with the increase in number of dimensions, so that the distance measurement between pairs of points becomes meaningless and the average density of points anywhere in the data is likely to be low.

### IV. GAP ANALYSIS

In today’s era the corporate and scientific environment produces massive amounts of data. To collect and analyze this data is a difficult task as data is increasing not in amount only but in complexity. Based on literature survey, there are various techniques which are used to analyze large datasets but these techniques are not efficient as some of them are related to particular task and do not provide the global solutions, some of them are fast but they had to compromise with the quality of clusters and vice versa. There has been lot of work done to improve efficiency of K-Means and DBSCAN algorithms to determine good quality clusters in less computation time but there are some shortcomings in both these techniques. Also there is a need to design new methodologies which can deal with the real time and online streaming data.

- The parallel k-means algorithm executes fast but it cannot handle non-arbitrary shape and also does not deal with the noisy data [8].
- The parallel DBSCAN algorithm can handle noise as well as non-arbitrary shape but it takes more computation time and is more complex than K-Means [9].
- The ELM feature is applied to K-Means to find accurate clusters in less execution time, the clusters
produced had to compromise in their quality also as large amount of resource is required to get an optimal solution [10].

- The K-Means is also applied with modified coherent intelligence which provides efficiency and reliability but in this boundary points are the problem [11].
- The DBCURE-MR is extension of DBSCAN which produces good quality clusters but it takes more computation time [12].

V. PROBLEM STATEMENT

Today big data has become buzz in the market. Among various challenges in analyzing big data the major issue is to design and develop the new techniques for clustering. Clustering techniques are used for analyzing big data in which cluster of similar objects are formed that is helpful for business world, weather forecasting etc. Cloud computing can be used for big data analysis but there is problem to analyze data on cloud environment as many traditional algorithms cannot be applied directly on cloud environment and also there is an issue of applying scalability on traditional algorithms, delay in result produced and accuracy of result produced. These issues can be addressed by K-Means and DBSCAN applied together. Therefore in this research work a parallel clustering algorithm is proposed and designed that helps in analyzing big data in an efficient manner.

A. Proposed Hybrid Technique

The problem stated is solved by a hybrid approach that is based on parallel k-means and parallel DBSCAN. It takes advantages of both algorithms and analyses big data in an efficient way. The proposed technique is considered whose purpose is to generate accurate clusters in less processing time.

The proposed method is a hybrid technique based on parallel K-Means and parallel DBSCAN that combines the benefits of both parallel K-Means and parallel DBSCAN algorithms. The benefit of parallel K-Means is that it is not complex and forms clusters in less execution time while the advantage of parallel DBSCAN is that it forms the cluster of arbitrary shape.

![Diagram](https://via.placeholder.com/150)

**Fig. 5.1:** Execution of Proposed Algorithm

Figure 5.1 shows that the main procedure of proposed method that works in three stages:

- Stage 1: The first stage is the data partition stage in which data is partitioned into different regions or parts to minimize the boundary points.
- Stage 2: In the second stage mapper function is performed on each node in which local clusters are formed.
- Stage 3: The third stage is the reducer stage in which mean of each cluster is calculated and it is returned as the cluster id of each cluster.

B. Proposed Algorithm

The design of proposed algorithms is as follows:

**Input:** D: Dataset having p data points.
- K: Number of clusters to be formed.
- eps: Radius to calculate neighborhood.
- minPts: Number of minimum points required to be in neighborhood.

**Output:** K clusters having Cm center mean as cluster id.

- **Step 1:** Partitioning of Data
  - The data points are sorted in ascending order.
  - The range of partition is defined by 2 eps.
  - Each partition gets its id as Pi.

- **Step 2:** Mapper Stage
  - Each partition gets its id as Pi.
  - Calculate Cm center mean of each cluster
  - Calculate total of data points in each cluster
  - Count the total number of clusters
  - Emit combination of partition id and cluster id PiCi as key and data points as value

- **Step 3:** Reducer Stage
  - Takes input from mapper stage as <key, value> pair where key is PiCi and value as data points.
  - Set counter as 0.
  - For each data point in partition region
  - If size of neighbor set <minPts
  - Mark data point as noise
  - Otherwise increment counter add data point to cluster
  - Emit combination of partition id and cluster id PiCi as key and data points as value

**VI. RESULT ANALYSIS**

To implement the proposed approach four different datasets are used. These datasets are collected from different repositories. The sample dataset is from Clustering Dataset Repository [21], US Census dataset is from UCI Machine Learning Repository [22], while Iris dataset and Balance dataset is from KEEL-dataset Repository [23]. All these datasets have different dimensionality. The proposed approach is implemented by Java using Eclipse on Hadoop.
platform under the environment of 2.3 GHz Intel(R) Core i3 processor, 8GB RAM and Ubuntu 12.0.4.

The experiments performed on these algorithms are on the basis of two parameters that is the execution time and accuracy performance.

**A. Test for Execution Time**

All the three algorithms that is K-Means-MR, DBSCAN-MR and proposed approach are tested on the same datasets to calculate their execution time. Table 6.1 shows the comparison of the execution time taken by each algorithm on four different datasets that includes sample dataset, US census dataset, iris dataset and balance scale dataset. Figure 6.2 shows the comparison graph of these algorithms for the all four datasets.

<table>
<thead>
<tr>
<th>Datasets</th>
<th>DBSCAN-MR</th>
<th>K-MEANS-MR</th>
<th>PROPOSED METHOD</th>
</tr>
</thead>
<tbody>
<tr>
<td>SAMPLE_DS</td>
<td>21.72</td>
<td>13.99</td>
<td>17.75</td>
</tr>
<tr>
<td>US_CENSUS</td>
<td>70.83</td>
<td>50.82</td>
<td>62.85</td>
</tr>
<tr>
<td>IRIS</td>
<td>15.54</td>
<td>9.29</td>
<td>12.71</td>
</tr>
<tr>
<td>BALANCE_SCALE</td>
<td>14.927</td>
<td>9.05</td>
<td>11.8</td>
</tr>
</tbody>
</table>

Table 6.1: Comparison of Execution Time (in seconds)

![Time Comparison Graph](image)

It is seen from graph that the proposed algorithm takes less execution time than DBSCAN-MR algorithm but when it compared to K-Means-MR it takes more time to execute because proposed algorithm has to deal with the noise while K-Means algorithm does not dealt with noise or boundary points. Noise is an important factor as it shows the uncertain behavior of the commodity which is helpful in detecting frauds cases. Thus proposed algorithm takes more execution time than K-Means-MR.

**B. Test for Performance of Accuracy**

All the three algorithms that is K-Means-MR, DBSCAN-MR and proposed approach are tested on two datasets that includes iris dataset and balance scale dataset to calculate the accuracy performance. Accuracy evaluation is needed to maintain the quality of clusters. The Random Index \[^{[17]}\] method is used to evaluate the performance of given algorithms. This is a commonly used method to measure similarity of two data clusters. The formula for Random Index is given below:

\[
R = \frac{a + b}{a + b + c + d} = \frac{(a + b)}{\binom{2}{2}}
\]

Where \( R \) is random index.

The Rand Index has a value between 0 and 1 where 0 indicates that two data clusters do not agree on any point of pairs while 1 indicate that clusters are exactly same.

<table>
<thead>
<tr>
<th>Datasets / Algorithms</th>
<th>DBSCAN-MR</th>
<th>K-MEANS-MR</th>
<th>PROPOSED METHOD</th>
</tr>
</thead>
<tbody>
<tr>
<td>IRIS</td>
<td>37%</td>
<td>49%</td>
<td>59%</td>
</tr>
<tr>
<td>BALANCE_SCALE</td>
<td>39%</td>
<td>48%</td>
<td>63%</td>
</tr>
</tbody>
</table>

Table 4.2: Performance Comparison of Accuracy

![Accuracy Comparison Graph](image)

When these algorithms are tested for the accuracy the proposed algorithm out performs both K-Means-MR and DBSCAN-MR. The performance graph and table clearly demonstrate that proposed algorithm performs better than both algorithms.

**VII. CONCLUSION**

The overall goal of data mining process is to extract the knowledge from large data. This thesis introduces big data and provides background of various clustering techniques used to analyze big data. In this work comparative analysis of these techniques is done. A hybrid approach based on parallel K-Means and parallel DBSCAN for efficient clustering of big data is proposed. This approach is developed in java, deployed in MapReduce framework of Hadoop. The experimental results have been gathered which shows that the proposed approach is more accurate as compared to MapReduce K-Means and MapReduce DBSCAN when tested on the four different datasets with different dimensions.

**VIII. LIMITATIONS**

The proposed approach still has some of the following limitations.

1. The proposed approach still requires the value of \( K \), the number of initial or desired clusters as input, though data points has been distributed.
2. ii) This approach can be applied only for those data sets which have numerical values or attributes.

**REFERENCES**


