Comparison of Enhanced DBSCAN Algorithms: A Review

Harsh Shinde
B.E. Student of Computer Engineering
Atharva College of Engineering, Mumbai University
Mumbai, MH, India

Amruta Sankhe
Prof. of Computer Engineering
Atharva College of Engineering, Mumbai University
Mumbai, MH, India

***Abstract*—While data containing any type of information is getting ubiquitous, proper processing of such data is very much obliged. To classify spatial data, various clustering algorithms have been invented. DBSCAN (Density Based Spatial Clustering of Application with Noise) is one of the consummate clustering algorithm with respect to discovery of arbitrarily shaped cluster in a spatial datasets. Due to its flexibility and tremendous research potential, DBSCAN algorithm is one of the most cited in scholarly literature. Considering the fact that most researchers tried to experiment and evaluate the algorithm, certain modifications of DBSCAN were made in order to have more efficient outcomes or to reduce time complexities. This paper discusses such modified algorithms and how they surpass the original DBSCAN in certain ways. Thorough analysis of each algorithm is mentioned and their critical evaluation is done accordingly. These algorithms are then comparatively evaluated with regards to various parameters.**

***Index Terms*—Clustering algorithms, DBSCAN, DDCAR, RDD-DBSCAN, FastDBSCAN, DDBSCAN, DSets-DBSCAN**

I. Introduction

Spatial data management needs proper evaluation and processing of any information to generate useful and desired data. For such implementation of data processing, the technique of Knowledge Discovery of Data (KDD), which is also known as Data Mining, is widely used. Clustering is a popularly used data mining process which classifies spatially represented datasets [1]. This classification of datasets results in formation of similar group of objects, possessing similar properties. To classify such type of data, a number of clustering algorithms have been invented and implemented, making classification of data into arbitrarily shaped, similar datasets called clusters. These clustering algorithms help to extract useful information generally in the form of patterns. Density-based clustering techniques are used to mine information containing large datasets.

Due to emerging trends of big data, density-based clustering algorithms have been mostly cited in scientific literature due to its tremendous research potential and also possessing vast choices of enhancement in its original algorithms. Although the characteristics of other algorithms might work well with similar datasets, density based clustering algorithms is more efficient with respect to any variation of datasets. Out of many density-based clustering algorithms such as DBSCAN, DENCLUE, DBCLASD, OPTICS, etc [8][9], DBSCAN (Density-Based Spatial Clustering of Application with Noise) is one of the most universally acclaimed. DBSCAN and other density-based algorithms are important because they are unaffected by noise points and can handle clusters of various shapes and sizes. They are a lot of clusters that DBSCAN can find that other algorithms would not be able to find. DBSCAN searches for core objects, i.e., objects having dense neighborhood. DBSCAN has a number of applications ranging from machine learning library to weather analysis or at atmospheric science on a national scale [3]. Due to its vast experimental potential, many researchers have cited and made some changes in the original algorithm. Some algorithm tries to reduce the computational process while other algorithm reduces time complexity over substantially varied datasets.

Researchers have been interested in DBSCAN since its proposal as this algorithm had some massive potential towards various data driven applications. Considering the emerging trends of big data and data mining tools, the applications of DBSCAN is also bringing effective results for various sets of data. DBSCAN possesses various limitations with respect to the variation and the size of datasets. Following such limitations, various advanced algorithms were invented for overcoming different types of shortcomings which the original DBSCAN possessed. These changes were made to enhance the restraints put forth by DBSCAN; some increase the effectiveness of the algorithm, while others produces similar results as the original algorithm but decreasing the time complexity taken by the DBSCAN [5].

The following paper gives an insight towards some recent changes made in the original DBSCAN. These changes are subjective to the overall development of the original algorithm and to overcome various drawbacks associated with DBSCAN. The algorithms discussed in this paper, providing profound understanding of each, are stated as follows: DBSCAN [2], DDCAR [3], RDD-DBSCAN [4], FastDBSCAN [5], DDBSCAN [6] and DSets-DBSCAN [7]. Each of the given algorithms is thoroughly analyzed and their critical evaluation is done accordingly. Moreover, a separate comparison is made with regards to certain parameters, to obtain a holistic view of the changes made by each of the algorithms.

The rest of the paper is organized as follows. We discuss the original DBSCAN algorithm in Section 2. In

Section 3, we present the summary and elaboration of different algorithms which provide certain modifications and improvements to the original DBSCAN algorithm. Section 4 provides the conclusion of our overall paper.

II. DBSCAN

DBSCAN (Density-Based Spatial Clustering of Application with Noise) is a density-based data clustering algorithm which was proposed by Martin Ester, Hans-Peter Kriegal, Jörg Sander and Xiaowei Xu in 1996 [2]. The main purpose of the algorithm is to connect core objects (objects having dense neighborhoods) and their neighbors to form dense regions as clusters. For a set of points in space, the algorithm groups together closely packed points and the points in the low density region are called outliers. Outliers can be used to detect irrelevant information, usually produced during fraud detection. This algorithm is flexible and dynamic with respect to data.

The steps of DBSCAN are given in the following points:

- For a point, make n-dimensional sphere of radius 'Eps', for n-dimensional datasets.
- Count the number of data points within the sphere. Indicate the value as 'p'.
- If p>min_pts, min_pts being the minimum points that should be present within the radius Eps, mark the center point to be a part of the cluster; also mark points inside Eps as a part of the cluster.
- Repeat this step to the other points in the sphere except the center, to expand the cluster.
- If p<min_pts, ignore the point p and proceed to another point in the dataset.

Following is a pseudo-code algorithm of DBSCAN [10]:

```
DBSCAN(D, eps, MinPts)
  { C = 0
  for each point P in dataset D
        { if P is visited
        continue next point
        mark P as visited
        NeighborPts = regionQuery(P,
        eps) if sizeof(NeighborPts) <
        MinPts mark P as NOISE
        else {
        C = next cluster
        expandCluster(P, NeighborPts, C, eps,
MinPts)
        }
  }
}
expandCluster(P, NeighborPts, C, eps, MinPts)
  { add P to cluster C
  for each point P' in NeighborPts {
        if P' is not visited
        { mark P' as
        visited
        NeighborPts' = regionQuery(P', eps)
        if sizeof(NeighborPts') >= MinPts
        NeighborPts = NeighborPts joined with
NeighborPts'
        }
        if P' is not yet member of any
        cluster add P' to cluster C
  }
}
regionQuery(P,
 eps){ return all points within
 P's eps
neighborhood (including P)
}
```

The overall average runtime complexity achieved by this algorithm is O(nlogn). The worst case runtime complexity is $O(n^2)$.

III. LITERATURE REVIEW

In this section, we will discuss and evaluate the different modifications of the original DBSCAN algorithm. Each of the following algorithms pinpoints various flaws on the implementation of DBSCAN algorithm. These flaws were then solved using certain approach towards each of the following algorithms. Complete evaluation of modified algorithms and relevant future work, if any, are also discussed. The following algorithms are the improved version of the DBSCAN:

DDCAR [3] (Data Density clustering using Automated Radii) is a fully autonomous data density based clustering technique. This technique was proposed by Richard Hyde, Plamen Angelov in 2014. The algorithm automatically determines the number of clusters and derives suitable initial radii. This algorithm is non iterative, i.e., it assigns each sample to the appropriate cluster only once. The main advantages of this study over DBSCAN [2] are:

- No prior knowledge of the number of cluster is required.
- No initial radii or other user input required.
- No knowledge of data density required.
- Assignment of data to their original cluster is done accurately.
- Time complexity is reduced and it is dependent on the clustering data.

RDD-DBSCAN [4] is an algorithm proposed by Irving Cordova and Teng-Sheng Moh in 2015. This algorithm addresses large datasets utility of DBSCAN as it is not efficient while working with Resilient Distributed Datasets, which are a fast data processing abstraction created directly for in-memory computation of large datasets. Experimentally, RDD-DBSCAN scales as the number of nodes in a given cluster scale, while generating the same results as the sequential version of DBSCAN. The main advantages of this study over DBSCAN [2] are:

- Overcoming the scalability limitations by operating in a fully distributed fashion.

- The algorithm scales as the number of nodes in a given cluster scales, while generating similar results as the sequential version of DBSCAN.
- Improving in-computation memory and is more efficient.

Future work can be done in the following areas:

- Loading complete datasets for a given partition into memory.
- Selection of better partitioning scheme for the data space.

FastDBSCAN [5] is proposed by Vu Viet Thang, D.V. Pantiukhin and A.I. Galushkin in 2015. This algorithm divides the data into k partitions (using k-means), then uses a min-max method to select points for DBSCAN clustering. It is a hybrid clustering algorithm as it uses a combination of k-means, min-max method and DBSCAN to recover the final clusters and outliers. The main advantages of this study over DBSCAN [2] are:

- Overcoming quadratic time complexity ($O(n^2)$) processed in DBSCAN.
- Improves computational time and accuracy.

Directions for further research are given as follows:

- Developing graph based clustering based on k-means.
- Applying the algorithm in intrusion detection system datasets.

DDBSCAN [6] (Different Densities-Based Spatial Clustering of Applications with Noise) is a modified version of DBSCAN [2]. It uses new concepts to deal with spatial datasets. This algorithm was proposed by M.F. Hassanin, M. Hassan and Abdalla Shoeb in 2015. The idea behind this algorithm is to define a density factor to the cluster and the object then define a threshold parameter as decision criteria to determine whether joining this object or not. On applying this technique, any cluster will contain indistinguishable density nodes. The main advantages of this study over DBSCAN [1] are:

- Multi-density cluster handling is enhanced.
- Effective clustering of adjacent clusters.
- Clustering of noise points amongst adjacent clusters.

Dsets-DBSCAN [7] is a parameter free hybrid clustering algorithm proposed by Jian Hou, Huijun Gao and Xuelong Li. This algorithm is mainly used in data clustering and image segmentation experiments. The algorithm functions in two main steps. First, we apply histogram equalization to similarity matrices before using it in clustering, to eliminate the regulation parameter. This makes the algorithm parameter-free. The next step is to Run Dsets clustering followed by clustering based on DBSCAN and extract cluster sequentially. The main advantages of this study over DBSCAN [2] are:

- Considering careful parameter tuning, this algorithm performs better than conventional DBSCAN.
- Efficiency is increased with regards to data clustering.

TABLE I. COMPARISON OF ENHANCED DBSCAN ALGORITHMS

Algorithm	DBSCAN	DDCAR	RDD-DBSCAN	FastDBSCAN	DDBSCAN	Dset-DBSCAN
Features	Grouping together closely packed points, called clusters.	Automatically determines the number of clusters and derives suitable data-driven radii by the use of recursive density equation.	A modified DBSCAN algorithm which takes full advantage of Apache Spark's parallel capabilities implemented in Scala programming language and run in top of Apache Spark.	Divides the data in 'k' partitioning (using k-means), then using a min-max method to select points for DBSCAN clustering.	Computing the density of a cluster with respect to radius value Eps and Min_pts. Then provide density threshold which is responsible for joining a point to a certain cluster or not.	Application of histogram equalization to pairwise similarity of input data. This makes the Dsets results independent of user specified parameters. Then extend clusters from Dsets with DBSCAN.
Advantages	Finds arbitrarily shaped clusters. Robust to outliers.	No prior knowledge of the number of clusters is required. No initial radii or other user input required. No knowledge of the density of the data is required. Fewer calculation by the use of recursive density equation.	Addresses large datasets utility of DBSCAN as it is not efficient while working with RDDs. Overcomes scalability issues of the traditional DBSCAN algorithm by operating in a fully distributed fashion. Efficient performance on	Overcomes quadratic time complexity processed in DBSCAN. Improves computational time. Improves clustering accuracy.	Overcomes the problems of: Multi-density cluster discovery. Adjacent cluster discovery. Noise points amongst clusters.	Parameter-free clustering algorithm. Prevents over-segmentation of arbitrary shape. Effective in data clustering.

		Efficiency is significant with larger datasets and big data.	Apache Spark platform. Improved memory management. Improved Efficiency.			
Disadvantages	Complex computational costs. Prior knowledge of the number of clusters is required. Knowledge of data density required. Scalability limitation. Multi-density cluster discovery. Noise points amongst adjacent clusters. Adjacent cluster discovery. Uses heavy user specified parameters.	In smaller datasets the time advantage may be reduced due carrying out the density calculation twice per cluster when adjusting the radii. Algorithm is non-iterative, assigns each sample to the appropriate cluster only once. Time complexity is dependent on the clustering data.	Limitation is the need to load the complete dataset for a given partition into memory. Selection of better partitioning scheme for the data space is unknown. Improvement on n-dimensional datasets (n>1) is unknown.	Needs more research to interpret these advantages of partitioning-based clustering and density-based clustering for constructing hybrid clustering.	-	Careful parameter tuning is required for optimum results.
Time complexity	Average runtime complexity : O(nlogn) Worst case runtime complexity : O(n^2) Non matrix based implementation complexity : O(n)	Varies. Depends on the size of the dataset.	O(n*logn)	Linear time complexity.	-	-
Parameters needed	2 (Eps and Min_pts)	None	3(Eps, Min_pts, Max_pts)	3(Input dataset D, number of cluster for k-means 'k', proportion of data 't')	3 (Eps,Min_pts,Density Threshold)	3 parameters for initial Histogram Equalization technique(Dataset 'D', Eps, Min_pts), none for Dsets+DBSCAN.
Application and uses	Fraud detection systems, data analysis.	Atmospheric sciences by using Hyper Pole to Pole Observations (HIPPO) datasets. Large climate based datasets.	Mainly used in Apache Spark.	Intrusion detection system by clustering. Experimenting on various datasets.	Simulation tasks. (Testing with real and artificial datasets.)	Data Clustering. Image segmentation

IV. CONCLUSION

In this paper we have discussed and compared different modified DBSCAN algorithms. The algorithms discussed were DBSCAN, DDCAR, RDD-DBSCAN, FastDBSCAN, DDBSCAN and DSets-DBSCAN. This evaluation was done due to many disadvantages which DBSCAN had regarding a number of its features. DDCAR and DSets-DBSCAN are parameter free clustering algorithms. They do not require a user input parameter. RDD-DBSCAN is effective while working with Resilient Distributed Datasets. FastDBSCAN improves computational time and accuracy by overcoming the quadratic time complexity possessed in the traditional DBSCAN. DDBSCAN combats the problem of multi-density clustering and generation of noise points amongst clusters. Thus we obtained a holistic view of the changes made by each of the algorithm.

REFERENCES

[1] Jiawei Han, Micheline Kamber and Jian Pei, "Data Mining: Concepts and Techniques, 3rd Edition", 2012.
[2] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in Proc. Int. Conf. Knowl. Discovery Data Mining, 1996, pages 226–231..
[3] Richard Hyde, Plamen Angelov, "A Fully Autonomous Data Density Based Clustering Technique" in Evolving and Autonomous Learning Systems (EALS), 2014 IEEE Symposium, Orlando, FL on 9-12 Dec. 2014,pages 116 - 123.
[4] Irving Cordova,Teng-Sheng Moh, "DBSCAN on Resilient Distributed Datasets" in High Performance Computing & Simulation (HPCS), 2015 International Conference, Amsterdam,20-24 July 2015,pages 531 - 540.
[5] Vu Viet Thang ,D. V. Pantiukhin ,A. I. Galushkin, "A Hybrid Clustering Algorithm: The FastDBSCAN" in 2015 International Conference on Engineering and Telecommunication (EnT), Moscow, 18-19 Nov. 2015, pages 69 - 74.
[6] Mohammad F. Hassanin,Mohamed Hassan,Abdalla Shoeb, "DDBSCAN: Different Densities-Based Spatial Clustering of Applications with Noise" in 2015 International Conference on Control, Instrumentation, Communication and Computational Technologies (ICCICCT), Kumaracoil, 18-19 Dec. 2015, pages 401 - 404.
[7] Jian Hou, Huijun Gao, Xuelong Li, "DSets-DBSCAN: A Parameter-Free Clustering Algorithm" in IEEE Transactions on Image Processing (Volume: 25, Issue: 7),2016, pages 3182 - 3193.
[8] Garima,Hina Gulati,P.K.Singh, "Clustering Techniques in Data Mining: A Comparison" in 2015 2nd International Conference on Computing for Sustainable Global Development (INDIACom),2015,Pages 410 - 415.
[9] Saif ur Rehman,Sohail Asghar,Simon Fong ,S. Sarasvady, "DBSCAN: Past, present and future" in Applications of Digital Information and Web Technologies (ICADIWT), 2014 Fifth International Conference,Bangalore,17-19 Feb. 2014, pages 232 - 238.
[10] Wikipedia. (2016, May, 27) DBSCAN from Wikipedia, the free encyclopedia. [Online] Available at: https://en.wikipedia.org/wiki/DBSCAN#Algorithm.