COMPARATIVE STUDY ON VARIOUS DENSITY BASED CLUSTERING AND ITS TYPES

[1]Akash Kulmitra, [2]Mr. Ram Nivas Giri
[1]Student, [2]Assistant Professor
[1]Computer Science and Engineering,
[1]Raipur Institute Of Technology, Raipur, India

Abstract— Clustering is a kind of unsupervised learning process in data mining and pattern recognition and most of the clustering algorithms are sensitive to their input parameters. So it is necessary to evaluate results of the clustering algorithms. But it is difficult to define which clustering designs are acceptable hence several clustering validation measures are developed. Clustering is used in various fields such as pattern recognition. In the following paper we will be studying about various density based scanning algorithms and there implementation on some of very common data sets with various output. Density based clustering methods are used for clustering spatial databases with noise. Density based Spatial Clustering of Applications with noise (DBSCAN) can discover clusters of arbitrary shapes and sizes effectively we will be seeing this in the following paper. We will be comparing DBSCAN algorithm with various density based scanning algorithms and will be calculating and the results with the datasets with noise.

Index Terms—DBSCAN, Clustering algorithm, Spatial Clustering, Unsupervised learning.

I. INTRODUCTION

Data clustering is the unsupervised learning technique which forms the given data into no of groups of subclasses such that the object of the same subclasses are more similar compared to the objects of the other subclasses. The technique of clustering is used in various fields such as image analysis [1] , pattern recognition[2], knowledge discovery[3] , and medical analysis[1] which posed to identify clusters with arbitrary shapes and sizes, determine input parameters of algorithms with min requirements of domain knowledge and still a good efficiency on large databases.

Various clustering methods follow various different approaches to solve a specific data set. One of the methods of clustering partitioned clustering uses k-means[4] algorithm to find clusters of spherical shapes only and need to supply no of clusters as an input to the algorithm. Kernel K-means detect arbitrary clusters by transforming them into kernel functions. Having a time complexity of $O(n^2)$ and hence not feasible for large data sets. Hierarchical clustering is the another method of clustering which partitions the data sets into hierarchical structure clusters are obtained by combining the subsets ate various levels using minimum distance criteria[5].The hierarchical method are having a time complexity of $O(n^2)$ and should also define an appropriate stopping condition for split and merge of partition for deriving a cluster. BIRCH[6], is a kind of hierarchical algorithm which uses tree based representation for reducing time complexity but it can only find spherical data sets and also clustering result is affected by input order of data. Currently, semi-supervised[7] and multi-view based[8][11] methods have shown effective improvement in the accuracy of clustering. Semi-supervised clustering algorithms utilize small amount of labeled data from the user for better clustering.

Density Based Spatial Clustering of Applications with Noise(DBSCAN)[9] is the initial lead of density based clustering algorithms which can discover arbitrary shapes and also handle noise outliners efficiently. DBSCAN has the quadratic time complexity with data set size which can be extended to large datasets by reducing its complexity using spatial index structures like R-trees[10] for finding neighbors of the pattern

Still, they can't be applied of the high dimensional data set. In the following paper we are showing the proposed DBSCAN algorithm with the comparison to its various density based clustering algorithms .and on evaluating we could see that the clustered results are similar to the output of the traditional DBSCAN algorithm but their running times are reduced.

Table 1 Notations

Symbol	Denotes
P	Input set of data patterns
I	A pattern in P
D	No of dimensions of a pattern
M	Size of data set

The following table1 consist of the notations used in the paper. While the reat of the paper is organized as follows Section 2 describes about the original DBSCAN method. Section 3 describes about the algorithms which are used with DBSCAN(Density based clustering of spatial data). Section 4 provides about the experimental analysis. Section 5 gives the conclusion and future scope.

II. RELATED WORK

Density based clustering methods can find arbitrary shaped clusters in the dataset and also insensitive to noise. In density based clustering methods clusters are formed by merging dense areas separated by regions of spares areas. DBSCAN is proposed for clustering large spatial databases with noise or outliers. OPTICS [11] is an extension DBSCAN which can find clusters with varying densities by creating an augmented ordering of given dataset representing a density-based cluster structure. This ordering is equal to a density-based clustering with varied range of parameter settings. H Gao.[12]proposed a parameter free clustering method that utilizes Affinity Propagation algorithm [13] to detect local densities in the dataset and obtains a normalized density list . Later, DBSCAN method is modified to cluster the dataset in terms of the parameters in the normalized density list. DENCLUE [8] defines clusters by a local maximum of estimated kernel

density function . A hill climbing procedure is used for assigning points to nearest local maximum. L-DBSCAN [14] is a hybrid density based clustering method that first derives a set of prototypes from the dataset using leaders clustering method[15] and runs DBSCAN on the prototypes to find clusters. Further, Rough-DBSCAN [16] is proposed by applying rough-set theory [17] to L-DBSCAN method It has a time complexity of O(n) but the cluster results are influenced by threshold parameter that is specified to derive the prototypes . Recently, fast and scalable density based clustering method using graphics processing units(GPU) are proposed to improve the performance DBSCAN[18]. M Tang(2016) [19] parallel and distributed versions of DBSCAN are proposed for handling large datasets using graph algorithmic concepts and achieves well balanced workload by taking advantages of tree based bottom-up approach to construct clusters.

2.1 DBSCAN: A density based approach

DBSCAN algorithm defines cluster as a region of densely connected points separated by regions of non-dense points. If similarity measure is taken as Euclidean distance the region is a hyper sphere of radius eps at the given point c as center.

- eps-neighborhood: for a point i ∈ P , the eps-neighborhood denotes the set of points whose distance from i is less than or equal to eps. The cardinality of eps-neighborhood defines the threshold density of i.
- eps-connected: for a pair of points i; $y \in P$, if $\|i - y\| \leq eps$, then i ,y are eps-connected points.
- From the view of a DBSCAN method every point in the dataset will fall into either core point or border point. Further a border point can be either noise point or density connected point
- Core point : A point with threshold density greater than or equal to minpts.
- Border point : A point with threshold density less than minpts.
- Noise point : A point d is a noise point if the threshold density of p is less than minpts and all points in the eps-neighborhood of d are border points.
- Density connected point: A border point with at least one core point in its eps-neighborhood.

DBSCAN algorithm takes two parameters as an input eps and the minpts.eps which specifies the maximum distance neighborhood for the given point minpts is the minimum number of points required in the eps-neighborhood of a point of a cluster. Initially all points are marked unvisited

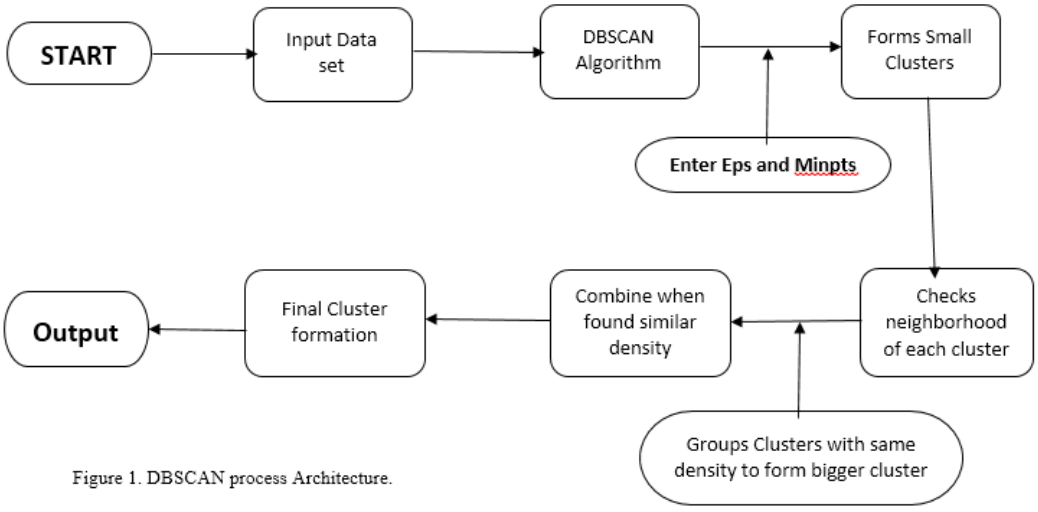

Figure 1. DBSCAN process Architecture.

The algorithm starts by randomly selecting an unvisited point and finding its eps-neighborhood. If the number of points in the its eps-neighborhood are less than the minpts then it is marked as noise or outliner ,otherwise it is considered as dense point and a new cluster is created . Further the same process is carried on is no new points can be added to the cluster then we see that the new cluster is complete and no points will be added to the cluster in subsequent iterations. And hence to find the new cluster from the given data set we have to repeat the same process up till the new cluster is formed. The process stops when all the nodes are allotted to some cluster or found as noise points. Every points in the cluster is eps-connected with at least one point in the same cluster to which it belongs and is not eps-connected with any other points in remaining clusters . However there may be points (border) which may be connected to the border points of other cluster, in that case the cluster is assigned to the cluster that processed it first. Such cases are rare in practice. Total no of eps-neighborhood operations performed is the size of the dataset.

G-DBSCAN :- G-DBSCAN[20] it is the improved version or model of the original DBSCAN algorithm. The main purpose behind this algorithm is that to reduce the number of query object which is being used as a starting point in the traditional DBSCAN , and arrange the data into the formats of grid, with the center point of the data into the grid which is to be replaced all the grid points as the input to the algorithm. As a result the query object will be reduced exponentially , which will go on to the improved and increased efficiency of the algorithm, and will result in the reduction of the memory footprints. But while solving the G-DBSCAN we should also know about the following things or parameters.

1. The size of the grid which is to be taken which taking the data as an input at the actual performance time.
2. The value of the noise threshold value so that the data which lies below the value of the threshold value will come under noise.
3. The center of the data into the grid.
4. Formula for distance calculation.

$$D = \sqrt{\sum_{i=0}^{n}(x_i - x)}$$

LSDBC(Locally Scaled Density Based Clustering algorithm)[21] :- Locally scaled density based clustering algorithm works by making cluster points by connecting dense regions of space until the density falls below a threshold determined by the center of the cluster. LSDBC takes two input parameters, k, the order of nearest neighbor to consider for each point in the dataset for density calculation and α, which determines the boundary of the current cluster expansion based on its density. The LSDBC algorithm first calculates the ε values for each point based on their kNN distances. ε allows us to order points based on their density. Smaller ε values correspond to denser regions in the dataset. The set of points are then sorted in ascending order of their ε. Algorithm 1 presents the main method of LSDBC. The function kNND istVal takes a point and a number k and returns the distance of the point to its kth nearest neighbor, ε, as well as the set of its k nearest neighbors. localMax function ensures that the selected point is the most dense point locally in its neighborhood.

OPticsXi[22]:- OPTICS can also be called as generalized DB clustering by creating an ordering of the points that allows the extraction of clusters with arbitrary values for ε .The generating-distance ε is the largest distance considered for clusters. Clusters can be extracted for all $\varepsilon_i$such that $0 \leq \varepsilon_i \leq \varepsilon$. The core-distance is the smallest distance ε' between p and an object in its ε-neighborhood such that p would be a core object. The reachability-distance of p is the smallest distance such that p is density-reachable from a core object o.

III. EXPERIMENT RESULT

In this section we present the experimental results performed to demonstrate the effectiveness of the algorithms. In this we performed the experiments on data sets obtained from UCI machine learning repository of varied dimensions and two dimensional synthetic data set (fig 1). In the second part we applied the various density based algorithms on the similar data to identify the total working time of the algorithms .In both the cases the performances of all the algorithms are evaluated.

All the experiments were performed on an Intel core i3 5th generation processor with 2.5GHz and 4GB RAM running windows 8.1. All programs are compiled and executed on the ELKI 0.7.1 (2016 February 11) GUI framework.

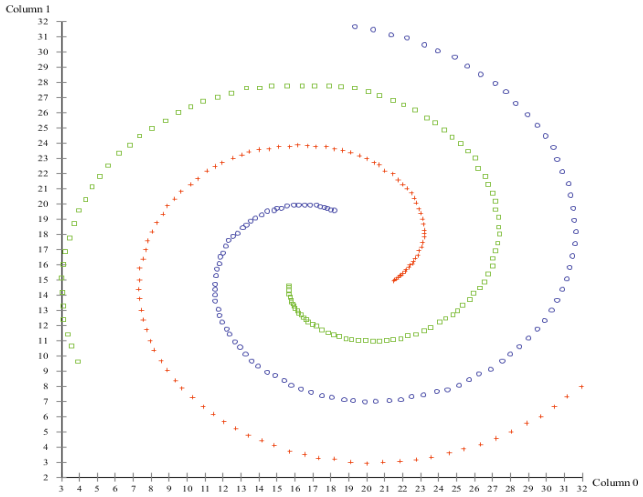

Figure 1. Synthetic dataset.

3.1 Experiment 1

In this study we had used six popular data sets from the UCI machine learning repository and one synthetic data set. The synthetic data sets are selected on the criteria of its corresponding dimension and size. Page blocks classification is composed of 5473 blocks obtained from 50 distinct documents. The blocks of the page layout of a document are obtained using segmentation process. The blocks are separate text from graphic areas. Letter recognition consist of database of image features used to identify 26 aplhabets.. synthetic data is manually generated with 3 classes of 4000 points each. A summary of data sets used is given in table no 2.

Table 2 Characteristics of datasets used in testing

Dataset	Size	Dimensions
Page blocks	6479	4
Letter recognition	20000	16
Image segmentation	2310	19
Satellite	6435	36
Plant species	1600	64
Concentric rings	1200	2
Iris	1200	4

Table 3 Running time comparisons of dataset for various algorithms. In (ms)

Dataset	G-DBSCAN	LSDBC	OpticsList	Opticsxi	OpticsHeap	FastOptics	DBScan
Page Blocks	10688	16001	18339	18650	19585	10172	1024
Letter recognition	303673	287914	286658	380824	287674	267673	146300
Image segmentation	31	63	31	36	31	0	221
Satellite	27250	31299	23700	217071	21545	47	296700
Plant species	5875	6354	4710	4032	4008	4023	33480
Iris	16	31	16	10	10	250	260

A performance comparison of proposed methods is shown in table 3. From the experimental results, it is observed that these algorithms perform well for data sets with dimensionality less than 20 and tend to perform poor as the dimensionality increarses.
OPTICS[11] uses the points of the database are (linearly) ordered such that points which are spatially closest become neighbors in the ordering. Additionally, a special distance is stored for each point that represents the density that needs to be accepted for a cluster in order to have both points belong to the same cluster. This is represented as a dendogram.

3.2 Experiment 2

In this experiment we check the scalability for the datasets with noise. Noise plays a very important role in the evaluation of the datasets as due to noise effect the efficiency of the algorithm may effect and this may also force change in the output.The checked results of all the density based algorithms is shown in the table4 with various parameters. The parameters which are being used to judge the efficiency and the working of the algorithms are as follows

- Jaccard Index.
- ARI (Adjusted Rand Index)
- Rand Index.
- F-measure
- Recall

Jaccard Index:- :- Jaccard similarity coefficient (Paul Jaccard), is a statistic used for comparing the similarity and diversity of sample sets. The Jaccard coefficient measures similarity between finite sample sets, and is defined as the size of the intersection divided by the size of the union of the sample sets

$$\mathrm{J}(\mathrm{A},\mathrm{B}) = \frac{|A \cap B|}{|A \cup B|} = \frac{|A \cap B|}{|A|+|B|-|A \cup B|}$$

(If A and B are both empty, we define $J(A,B) = 1$.)
The Jaccard distance, which measures *dis*similarity between sample sets, is complementary to the Jaccard coefficient and is obtained by subtracting the Jaccard coefficient from 1, or, equivalently, by dividing the difference of the sizes of the union and the intersection of two sets by the size of the union

$$d_j(\mathrm{A},\mathrm{B}) = 1 - \mathrm{J}(\mathrm{A},\mathrm{B}) \frac{|A \cup B| - |A \cap B|}{|A \cup B|}$$

This distance is a metric on the collection of all finite sets[23].
Adjusted Rand Index (ARI) :- The adjusted Rand index is the corrected-for-chance version of the Rand index.[24] Though the Rand Index may only yield a value between 0 and +1, the adjusted Rand index can yield negative values if the index is less than the expected index.[25]. The adjusted form of the Rand Index, the Adjusted Rand Index, is

$$AdjustedIndex = \frac{Index - ExpectedIndex}{MaxIndex - ExpectedIndex}$$

Rand Index :- The Rand index[24] or Rand measure (named after William M. Rand) in statistics, and in particular in data clustering, is a measure of the similarity between two data clusterings. A form of the Rand index may be defined that is adjusted for the chance grouping of elements, this is the adjusted Rand index. From a mathematical standpoint, Rand index is related to the accuracy, but is applicable even when class labels are not used.
F-mesaure :- It is used to balance the contribution of False negatives by weighting recall through a parameter β>0.
Where

$$\mathrm{P} = \frac{TP}{TP+FP},$$
$$\mathrm{R} = \frac{TP}{TP+FN}$$

Where,
P= precision and R= recall, F-measure can be represented as

$$F_\beta = (\beta\ \hat{}2 + 1).P.R/\beta\ \hat{}2\ (P+R)$$

Table 4 Evaluation of DBSCAN:

Index values	Satellite	Plant
Jaccard	0.7973	0.9970
ARI	0.337	0.4985
Rand	0.7974	0.9970
F-measure	0.8872	0.9985
Recall	0.7973	0.9985

Evaluation of GDBScan

Index values	Satellite	Plant
Jaccard	0.7973	0.3845
ARI	0.3376	0.2933
Rand	0.7974	0.98
F-measure	0.8872	0.7413
Recall	0.7973	1.0

Evaluation of LSDBC

Index values	Satellite	Plant
Jaccard	0.5422	0.3949
ARI	-3.800	0.5850
Rand	0.5423	0.8721
F-measure	0.7034	0.7598
Recall	0.5423	0.5924

Evaluation of OPTICSxi

Index values	Satellite	Plant
Jaccard	0.7717	0.3845
ARI	0.002	0.2933
Rand	0.7718	0.38
F-measure	0.8711	0.7413
Recall	0.7717	0.379

IV. CONCLUSION

In this paper we have studied different density based algorithms with respect to some popular datasets and we came to know that the results which we got from evaluating the following data sets is that there are some density based algorithms which are better and efficient than DBSCAN algorithm

REFERENCES

[1] Kutarnia and P. Pedersen, "A Markov random field approach to group-wise registration / mosaicing with application to ultrasound," Med. Image Anal., vol. 24, no. 1, pp. 106–124, 2015.

[2] D. Huang, J. Lai, and C. Wang, "Ensemble clustering using factor graph," Pattern Recognit., vol. 50, pp. 131–142, 2016.

[3] A. Hinneburg and D. Keim, "An Efficient Approach to Clustering in Large Multimedia Databases with Noise," Proc. 4th Int. Conf. Knowl. Discov. Data Min. (KDD 98), no. October 2001, pp. 58–65, 1998.

[4] G. Hamerly, "Making k -means even faster," pp. 130–140.

[5] X. Tang and P. Zhu, "Hierarchical Clustering Problems and Analysis of Fuzzy Proximity Relation on Granular Space," vol. 21, no. 5, pp. 814–824, 2013.

[6] C. Liabilities and P. Accounting, "Chapter 11," pp. 259–276, 2002.

[7] H. Zhang and J. Lu, "Knowledge-Based Systems Semi-supervised fuzzy clustering : A kernel-based approach," Knowledge-Based Syst., vol. 22, no. 6, pp. 477–481, 2009.

[8] H. Rehioui, A. Idrissi, M. Abourezq, and F. Zegrari, "DENCLUE-IM : A New Approach for Big Data Clustering," Procedia - Procedia Comput. Sci., vol. 83, no. Ant, pp. 560–567, 2016.

[9] L. Kieu, A. Bhaskar, and E. Chung, "A modified Density-Based Scanning Algorithm with Noise for spatial travel pattern analysis from Smart Card AFC data," Transp. Res. Part C, vol. 58, pp. 193–207, 2015.

[10] J. Kuan, "nearest neighbour search. for R-tree family," no. September, pp. 9–12, 1997.

[11] M. Ankerst, M. M. Breunig, and H. Kriegel, "OPTICS : Ordering Points To Identify the Clustering Structure," pp. 49–60, 1999.

[12] J. Hou, H. Gao, and X. Li, "DSets-DBSCAN : A Parameter-Free," vol. 25, no. 7, pp. 3182–3193, 2016.

[13] D. Xu and Y. Tian, "A Comprehensive Survey of Clustering Algorithms," Ann. Data Sci., vol. 2, no. 2, pp. 165–193, 2015.

[14] T. Ali, S. Asghar, N. A. Sajid, and M. Ali, "Critical Analysis of DBSCAN Variations," 2010.

[15] K. M. Kumar and A. R. M. Reddy, "Author ' s Accepted Manuscript," Pattern Recognit., 2016.

[16] P. Viswanath and V. S. Babu, "Rough -DBSCAN : A fast hybrid density based clustering method for large data sets," Pattern Recognit. Lett., vol. 30, no. 16, pp. 1477–1488, 2009.

IJNRD Research Through Innovation