

# COMPARISON OF DENSITY-BASED CLUSTERING ALGORITHMS

Mariam Rehman

*Lahore College for Women University  
Lahore, Pakistan*

Syed Atif Mehdi

*University of Management and Technology  
Lahore, Pakistan*

## ABSTRACT

Clustering, in data mining, is a useful technique for discovering interesting data distributions and patterns in the underlying data, and has many application fields, such as statistical data analysis, pattern recognition, image processing, and other business applications. Although researchers have been working on clustering algorithms for decades, and a lot of algorithms for clustering have been developed; there is still much work to be done to develop an algorithm or technique for clustering very large databases and high dimensional data. As an outstanding representative of clustering algorithms, DBSCAN algorithm shows good performance in spatial data clustering. In the research work, comparison between DBSCAN and RDBC algorithms has been presented.

## KEYWORDS

Density based spatial clustering of applications with noise (DBSCAN), Recursive density based clustering (RDBC)

## 1. INTRODUCTION

The objective of the research work is to determine the usefulness of data mining in various environments, detailed study of density based algorithms (DBSCAN, RDBC) and also to provide comparison between density-based algorithms by implementing the said algorithms.

## 2. BACKGROUND STUDY

Density based algorithms typically regard clusters as dense regions of objects in the data space that are separated by regions of low density. The main idea of density-based approach is to find regions of high-density and low density, with high-density regions being separated from low-density regions. These approaches can make it easy to discover arbitrary clusters. A common way is to divide the high-dimensional space into density-based grid units. Units containing relatively high densities are the cluster centers and the boundaries between clusters fall in the regions of low-density units. (Bradley & Fayyad, 1998)

There are different algorithms for clustering. Some of the clustering algorithms require that the number of clusters should be known prior to the start of clustering process others determine the clusters themselves usually Density-based clustering algorithms are independent of prior knowledge of number of cluster. Such algorithms may be useful in situations where the number of cluster should be determined easily before the start of the algorithm (Han & Kamber, 2000).

## 2.1 Density based Spatial Clustering of Applications with Noise (DBSCAN)

Density based spatial clustering of applications with noise, DBSCAN; rely on a density-based notion of clusters, which is designed to discover clusters of arbitrary shape and also have ability to handle noise. The main task of this algorithm is class identification, i.e. the grouping of the objects into meaningful subclasses.

Two global parameters for DBSCAN algorithms are:

- Eps: Maximum radius of the neighborhood
- MinPts: Minimum number of points in an Eps-neighborhood of that point

*Core Object*: Object with at least MinPts objects within a radius 'Eps-neighborhood'

*Border Object*: Object that on the border of a cluster

$NEps(p): \{q \text{ belongs to } D \mid dist(p,q) \leq Eps\}$

*Directly Density-Reachable*: A point  $p$  is directly density-reachable from a point  $q$  w.r.t  $Eps$ ,  $MinPts$  if

$p \text{ belongs to } NEps(q)$

$|NEps(q)| \geq MinPts$

*Density-Reachable*: A point  $p$  is density-reachable from a point  $q$  w.r.t  $Eps$ ,  $MinPts$  if there is a chain of points  $p_1, \dots, p_n, p_1 = q, p_n = p$  such that  $p_{i+1}$  is directly density-reachable from  $p_i$

*Density-Connected*: A point  $p$  is density-connected to a point  $q$  w.r.t  $Eps$ ,  $MinPts$  if there is a point  $o$  such that both,  $p$  and  $q$  are density-reachable from  $o$  w.r.t  $Eps$  and  $MinPts$ .

*Algorithm*: The algorithm of DBSCAN is as follows (M. Ester, H. P. Kriegel, J. Sander, 1996)

- Arbitrary select a point  $p$
- Retrieve all points density-reachable from  $p$  w.r.t  $Eps$  and  $MinPts$ .
- If  $p$  is a core point, a cluster is formed.
- If  $p$  is a border point, no points are density-reachable from  $p$  and DBSCAN visits the next point of the database.
- Continue the process until all of the points have been processed.

## 2.2 Recursive Density based Clustering Algorithm (RDBC)

RDBC is an extended form of DBSCAN, an algorithm to group neighboring objects of the database into clusters. Moreover, it does not require a predetermined cluster number to operate (Zhong Su, Qiang Yang, Hongjiang Zhang, Xiaowei Xu, Yuhon Hu, 1999).

The algorithm is based on DBSCAN and is applicable to any database containing data from a metric space, e.g., to a web log database. Clustering algorithm calculates a density measure based on the distance metrics that is computed from the data set according to the distance definition. It then selects the points that are dense enough in the space of distance metrics and constructs an abstract space based on these points. It does this recursively until no more abstraction space can be built because it can change the parameters intelligently during the recursively process, RDBC can yield results superior than that of DBSCAN which will be proved in experiments.

RDBC is an improvement of DBSCAN. In RDBC, it calls DBSCAN with different distance thresholds  $\epsilon$  and density threshold  $MinPts$ , and returns the result when the number of clusters is appropriate. The key difference between RDBC and DBSCAN is that in RDBC, the identification of core points are performed separately from that of clustering each individual data points. This is called an abstraction because these core points can be regarded as clustering centers that are representative of the data points. For this purpose, different values of  $\epsilon$  and  $MinPts$  have been used in RDBC to identify this core point set, **CSet**. Only after appropriate **CSet** is determined, the core points are clustered, and the remaining data points are then assigned to clusters according to their proximity to a particular cluster.

### 2.2.1 RDBC Algorithm

Set initial values  $\epsilon = \epsilon_1$  and  $Mpts = Mpts_1$

DataSet = data\_set;

RDBC ( $\epsilon$ ,  $Mpts$ , DataSet)

Use  $\epsilon$  and  $Mpts$  to get the core points set CSet

If size (CSet) > size (DataSet) / 2 // Stopping criteria is met.

```

DBSCAN (DataSet,  $\epsilon$ , Mpts);
Else // Continue to abstract core points;
 $\epsilon = \epsilon / 2$ ; Mpts = Mpts / 4
RDBC ( $\epsilon$ , Mpts, CSet); // Collect all other points in around clusters

```

The algorithm goes into a cycle in which the core points themselves are taken as the points in a space, and clustering is done on those core points with stricter requirement on a core-point (with smaller radius around a core point). This process stops when nearly half the points that remain are core points. Then, the algorithm will begin a gathering process to gather the rest of the points around the core points found into clusters. This is done with a larger radius value  $\epsilon^2$ .

In particular, following steps have been executed for RDBC algorithm.

- Use pre-defined values of  $\epsilon$  and MinPts to compute core points and also place them in one variable.
- Perform DBSCAN on the variable to cluster core points only;
- Assign remaining data points not in variable to the clusters formed by core points.

### 3. COMPARISON

This section of paper offers an overview of the main characteristics of the clustering algorithms presented in a comparative way. Table 1 summarizes the main concepts and the characteristics of the DBSCAN and RDBC algorithms. Study is based on the following features of the algorithms:

- Complexity
- Input Parameters
- The type of the data that an algorithm supports (numerical, categorical)
- The shape of clusters
- Ability to handle noise and outliers
- Clustering Results
- The Clustering Criterion

In the research work, study is being done on the DBSCAN and RDBC clustering algorithms. They suitably handle arbitrary shaped collections of points (e.g. ellipsoidal, spiral, and cylindrical) as well as clusters of different sizes. Moreover, they can efficiently separate noise (outliers).

Table 1. Comparison between DBSCAN and RDBC

Name	Complexity	Input Parameters	Geometry	Noise	Results	Clustering Criterion
DBSCAN	$O(n^2)$	Cluster Radius, Minimum number of Objects	Arbitrary Shapes	Yes	Assignment of data values to clusters	Merge points that are density-reachable into one cluster
RDBC	$O(n^2)$	Intelligent parameter settings (initially we define some value then it goes recursively)	Arbitrary Shapes	Yes	Identification of Core Points is performed separately	Clustering is done on Core Points

### 4. EXPERIMENTS AND RESULTS

Firstly, the algorithms have been implemented and tested using the well-known benchmark iris data set. Perform testing of the application by the use of algorithms DBSCAN, and RDBC, and finally made comparison between both of these algorithms on the same dataset has also been done. The benefit of using the benchmark data is that it confirmed the validity of the experiments.

#### 4.1 Iris Problem

Iris data consists of classifying three species of flowers, setosa, versicolor and virginica. There are 150

samples of this problem, 50 of each class. A sample consists of four dimensions representing the four attributes of iris flower. The attributes are sepal length, sepal width, and petal length and petal width.

## 4.2 Experiment and Results (Iris Problem)

The following results have been obtained by applying DBSCAN and RDBC on the Iris data set.

Table 2: Comparing clusters obtained by RDBC and DBSCAN on Iris Data

Sr. No	Eps	Min Point	No. of Clusters (DBSCAN)	No. of Clusters (RDBC)
1	1.5	3	5	5
2	2	3	5	6
3	2.5	3	7	7
4	3	3	10	12
5	3.5	3	10	12

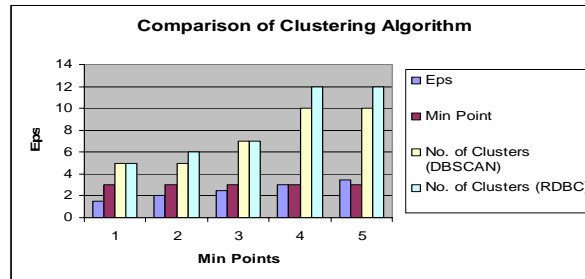


Figure 4.2.1: Graphical

Representations of Results

Table 2 shows the clustering result and comparison between the two clustering algorithms. RDBC have the same time complexity as that of DBSCAN. More clusters have been obtained in RDBC.

## 4.3 Flying Fitness Problem

Flying Fitness data consists of classifying six variables. There are 40 samples of this problem, 40 for each class. A sample consists of six dimensions representing the six attributes.

## 4.4 Experiment and Results (Flying Fitness Problem)

The following results have been obtained by applying DBSCAN and RDBC on the Flying Fitness data set.

Table 3: Comparing clusters obtained by RDBC and DBSCAN on Flying Fitness Data Set

Sr. No	Eps	Min Point	No. of Clusters (DBSCAN)	No. of Clusters (RDBC)
1	0.5	3	8	8
2	1	3	8	8
3	0.5	4	5	7
4	1	4	5	7

## 4.5 Performance Evaluation

DBSCAN constructs clusters using distance transitivity based on a density measure defined by the user. DBSCAN performs this clustering using a fixed threshold value to determine “dense” regions. Because this

threshold value is constant across all points in the space, that's why the algorithm often cannot distinguish between dense and loose points, and as a consequence often the whole data is lumped into a single cluster. To remedy this problem, RDBC has been used that attempts to solve this problem by varying  $\epsilon$  and MinPts whenever necessary.

The runtime comparison of DBSCAN and RDBC on the Iris Data Set is shown below which is almost the same. We see that using RDBC, while having about the same time complexity as DBSCAN, we obtain more clusters (as shown in Figure 4.2.1) for the data set that is more reasonable and will generate clusters with more even distribution than DBSCAN.

Table 3: Performance Evaluation

Sr. No	Content	RDBC	DBSCAN
1	Number of Data Points	150	150
2	Run Time	$O(n^2) / 3 \text{ Sec}$	$O(n^2) / 4 \text{ Sec}$

More clusters are good because they are able to separate noise while generation of clusters. RDBC have more ability than DBSCAN to handle noise as RDBC is generating more clusters.

## 5. CONCLUSION

Clustering algorithms are attractive for the task of class identification in spatial databases. In this work, focus has been made over the comparison of clustering algorithms i.e. DBSCAN and RDBC. DBSCAN relies on a density-based notion of clusters. It requires only one input parameter and supports the user in determining an appropriate value for it while RDBC perform intelligent parameter settings.

Performance evaluation was performed on Iris dataset. The results of these experiments demonstrate that RDBC is significantly more effective in discovering clusters of arbitrary shape than the well-known algorithm DBSCAN. Furthermore, the experiments have shown that RDBC is superior to that of DBSCAN and hence generating more clusters. When DBSCAN is applied to data set, it has been observed that fixed values of  $\epsilon$  and MinPts often leads to a single, giant cluster which is not useful at all. The remedy of problem, algorithm has been used called RDBC that attempts to solve the problem of bigger clusters by varying  $\epsilon$  and MinPts whenever necessary.

## ACKNOWLEDGEMENT

We are thankful to Mr. Ahmar Mirza for his valuable advices and guidance throughout the research work.

## REFERENCES

- Bradley PS, Fayyad U, Reinna C, 1998, scaling clustering algorithms to large databases.
- Han, Jiawei & Kamber, Micheline, Data Mining: Concepts and Techniques, San Francisco: Morgan Kaufmann Publisher 2000.
- Rakesh Agrawal, Johannes Gehrke, Dimitrios Gunopulos and Prabhakar Raghavan, 1998, Automatic Subspace Clustering for High Dimensional Data for Data Mining Application. *In Proceedings ACM SIGMOD International Conference on Management of Data, Seattle, Washington, USA*, pp. 94-105.
- M. Ester, H. P. Kriegel, J. Sander, X. Xu, 1996, A density – based algorithm for discovering clusters in large spatial databases with noise, *Proceedings of 2<sup>nd</sup> International Conference of Knowledge Discovery and Data Mining*.
- Mihael Ankerst, Markus M, Breunig, Hans-Peter Kriegel, Jorg Sander, 1999 *Proc. ACM SIGMOD' 99 Int. Conf. on Management of Data*, Philadelphia, (PA 1999).
- R. Ng and J. Han, 1994, Efficient and Effective Clustering Methods for Data Mining. *Proc. Of 1994 Int'l Conf. On Very Large Data Bases (VLDB'94)*, Santiago, Chile, pp 144-155.
- Zhong Su, Qiang Yang, Hongjiang Zhang, Xiaowei Xu, Yuhua Hu , 1999, Correlation-based Document Clustering using Web Logs *Department of Computing Science, Tsinghua University, Beijing 100084, China*