IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661,p-ISSN: 2278-8727, Volume 20, Issue 3, Ver. I (May. - June. 2018), PP 78-82
www.iosrjournals.org

The Bisecting Min Max DBSCAN Algorithm

Dr. Terence Johnson[1], Kedar Prabhu[2], Shubham Parvatkar[3], Apurva Naik[4], Pundalik Temkar[5]

[1]*(Associate Professor, Dept. of MCA, Finolex Academy of Management and Technology, Ratnagiri, India)*
[2]*(Software Developer, Open Destination Infotech, Goa, India)*
[3]*(Software Developer, Tentwenty, Goa, India)*
[4]*(Freelance Software Developer, Goa, India)*
[5]*(Freelance Software Developer, Goa, India)*
Corresponding Author: Dr. Terence Johnson

Abstract : *DBSCAN is a density based clustering algorithm which groups together data points of similar characteristics. It is based on two input parameters minpoints and epsilon. The disadvantage of DBSCAN algorithm is that it compares each point in the dataset with every other point in the dataset and resulting in runtime complexity of O(n^2). In this paper, a new approach of finding clusters similar to the clusters formed by DBSCAN but with improved time complexity is introduced. A quantitative performance analysis of the new methodology, the Bisecting Min Max DBSCAN algorithm on the iris dataset proved that, not only is the algorithm faster than the traditional DBSCAN, but also gives good cluster quality.*

Keywords - *Bisecting Min Max Clustering, Clustering, Data Mining, DBSCAN, Min Max Clustering*

Date of Submission: 14-05-2018 Date of acceptance: 31-05-2018

I. Introduction

Clustering is an important part in data mining in which we group together data objects with similar characteristics into groups called clusters where each cluster has different characteristics than the other clusters. DBSCAN is one such clustering algorithm in which data points are grouped together based on the proximities of the data points. In DBSCAN we input two parameters Min Points (Minpts) which represents the minimum number of points to form a cluster and Epsilon (Eps) which indicates the minimum distance required for a point to be a part of the cluster. A data point can be categorized as core point, border point or noise point. A point is a core point if the number of points within its Eps are more than Minpts. A point is a border point if the number of points within its Eps are less than the Minpts but it is in the neighborhood of core point. A noise point is any point that does not belong to above categories. Some of the important concepts used in DBSCAN are as follows:

Directly Density Reachability: Two points a and b are said to be density reachable if a is a core point and b lies in a’s Eps-neighborhood.

Density-Connected: a and b are density-connected if they are commonly density-reachable from another point o.

The advantages of DBSCAN is that it can form arbitrary shaped clusters, detect outliers and the number of clusters to be formed may not be specified at the beginning. One of the disadvantages of DBSCAN is that the two input parameters, Minpts and epsilon should be carefully set or done by a domain expert but the major disadvantage of DBSCAN is the runtime complexity. Since the algorithm is such that each data point in the dataset compares with every other data point in the dataset the time complexity becomes O(n^2).

II. Main Idea Of The Proposed Clustering Algorithm

To form a cluster, The Bisecting Min Max DBSCAN Algorithm requires 2 input parameters: Epsilon and Minpts . It first calculates the minimum point of the dataset from the origin and then calculates maximum from the minimum.

The following gives the complete illustration of the algorithm.

```
Let X= {x1,x2,x3….xn} be the sets of data points.
//Requires 2 parameters: E(eps) and minimum number of points
required to form a cluster(minpts).
Size(dataset)=Number of points in dataset
Step 1: Input 2 parameters ,E(eps) and Minpts.
Step 2: MinMaxClustering(dataset)
                    Find min and max points of the dataset.
Step 3:  if d(min,max) <= eps
             if size(dataset)>=minpts
                   form cluster
                              mark as visited
             else
                  mark all the points as NOISE

else if d(min,max)>=2(eps)+1
             MinMaxBisection(min,max,dataset)
             MinMaxClustering(dataset1)
             MinMaxClustering(dataset2)
 else
 {
   for each point  P in dataset
      if P is visited
          continue with  the next point
      neighborpts=getNeighborpts(P,eps)
          if size-of(neighborpts)<Minpts
           mark P as NOISE
      else
          DensityReachability(P,neighborpts,eps,Minpts)
           Mark partition as visited
}
DensityReachability(P,neighborpts,eps,Minpts)
{
   add P to cluster
   for each point P' in neighborpts
  {    if P' is not visited
      {
          mark P' as visited
         neighborpts'=getNeighborpts(P',eps)
         if size(neighborpts')>=minpts
               neighborpts+=neighborpts'
      }
  }
   if P' is not a yet a member of any cluster add P' to cluster
 }
MinMaxBisection(min,max,dataset)
{
   Create two partitions as dataset1 and dataset 2
   for each element P in dataset
     if (d(min,P)<=d(max,P))
         dataset1.append (P)
     else
         dataset2.append(p)
}
getNeighborpts(P,eps)
{
   return neighbors of point P
}
```

Min and max are the minimum and maximum points of the dataset respectively.We make use of Euclidean distance formula to compute distance between two points. Eps and minpts are the input parameters whose values are determined manually or depends on the application.

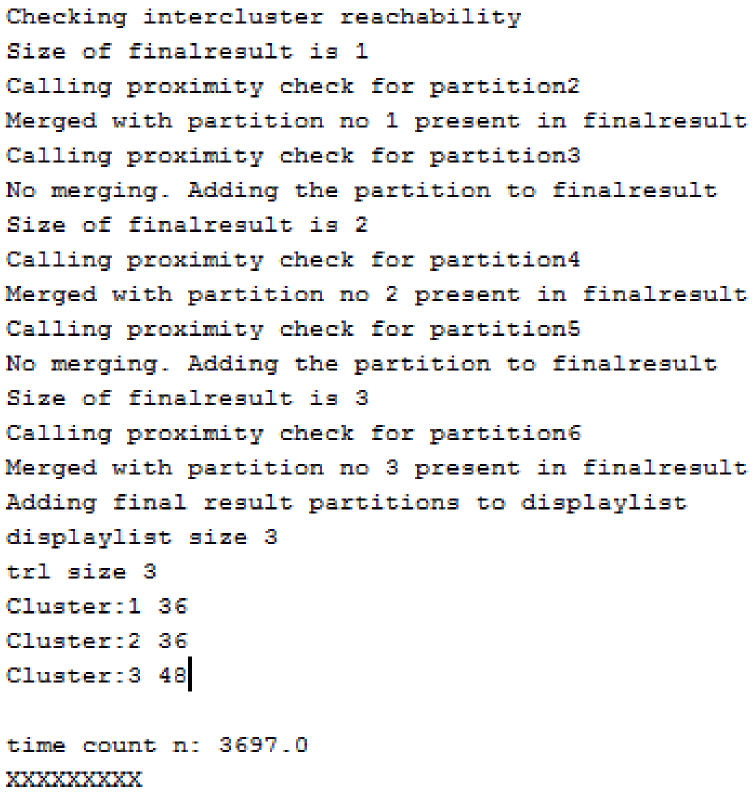
The function MinMaxBisection bisects the dataset into 2 partitions: Min partition and Max partition.

Based on the Euclidean distance from the minimum and maximum points, it assigns data points to these partitions.

getNeighborpts() returns all the points lying within the eps region of point under consideration.

III. Implementation And Results

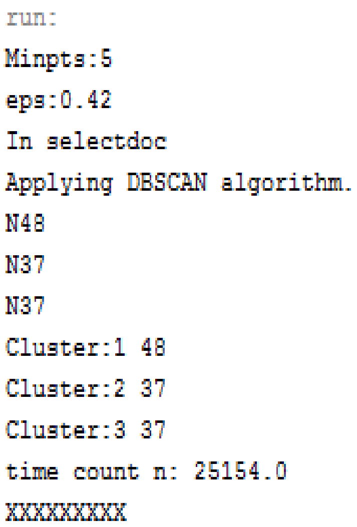
The algorithms were tested with Iris Dataset. The Iris dataset consists of 150 datapoints with 3 different classes(Setosa, Versicolor, Virginca). Each datapoint has four attributes (Sepal length, Sepal width, Petal length, Petal width). On implementing DBSCAN and The Bisecting Min Max DBSCAN Algorithm on iris dataset the following result is obtained.



```
Checking intercluster reachability
Size of finalresult is 1
Calling proximity check for partition2
Merged with partition no 1 present in finalresult
Calling proximity check for partition3
No merging. Adding the partition to finalresult
Size of finalresult is 2
Calling proximity check for partition4
Merged with partition no 2 present in finalresult
Calling proximity check for partition5
No merging. Adding the partition to finalresult
Size of finalresult is 3
Calling proximity check for partition6
Merged with partition no 3 present in finalresult
Adding final result partitions to displaylist
displaylist size 3
trl size 3
Cluster:1 36
Cluster:2 36
Cluster:3 48

time count n: 3697.0
XXXXXXXXX
```

Fig1: Output Of Bisecting Min Max DBSCAN Algorithm For Iris Dataset



```
run:
Minpts:5
eps:0.42
In selectdoc
Applying DBSCAN algorithm.
N48
N37
N37
Cluster:1 48
Cluster:2 37
Cluster:3 37
time count n: 25154.0
XXXXXXXXX
```

Fig2: Output of DBSCAN Algorithm for iris dataset

The output clearly shows that

a) the clusters formed in bisecting Min-Max DBSCAN Algorithm are similar to that of DBSCAN Algorithm.
b) The number of iterations executed in Bisecting Min Max DBSCAN Algorithm(3697) are much less than that of DBSCAN Algorithm(25154).

IV. Quantitative Performance Analysis

A cluster is said to have datapoints with similar characteristics where each cluster differs from another. We performed quantitative performance analysis of Iris dataset using DBSCAN and The Bisecting Min Max DBSCAN Algorithm. Precision, Recall, F-Measure, Accuracy and Error Rate are some of the metrics used in the analysis.

The results obtained for DBSCAN are as follows:

TABLE 1. Results for DBSCAN

DBSCAN				
Attributes	TP	TN	FP	FN
Iris Setosa	48	100	0	2
Iris Versicolor	37	100	0	13
Iris Virginica	33	96	4	17

TABLE 2. Performance Metrics for DBSCAN

DBSCAN			
Metrics	Iris Setosa	Iris Versicolor	Iris Virginica
Precision	100	100	89.18
Recall	96	74	66
F-measure	97.95	85.05	75.86
Accuracy	98.66	91.33	86
Error Rate	1.33	8.66	14

The results obtained for the Bisecting Min Max DBSCAN algorithm are as follows:

TABLE 3. Results for Bisecting Min Max DBSCAN Algorithm

The Bisecting Min Max DBSCAN Algorithm				
Attributes	TP	TN	FP	FN
Setosa	48	100	0	2
Versicolor	36	100	0	14
Virginica	32	96	4	18

TABLE 4. Performance Metrics for Bisecting Min Max DBSCAN Algorithm

The Bisecting Min Max DBSCAN Algorithm			
Metrics	Iris Setosa	Iris Versicolor	Iris Virginica
Precision	100	100	88.88
Recall	96	72	*64*
F-measure	97.95	83.72	74.41
Accuracy	98.66	90.66	85.33
Error Rate	1.33	9.33	14.66

The various performance parameters like precision, recall, F- Measure, Accuracy and Error Rate are computed using the following formulae

- **Precision** $\frac{TP}{A}$
- **Recall** $\frac{TP}{B}$
- **F-Measure** $\frac{2*Precision*Recall}{Precision + Recall}$
- **Accuracy** $\frac{C}{D}$
- **Error Rate** $\frac{E}{D}$

where

TP=True Positives, TN=True Negatives, FP=False Positives, FN=False Negatives

A=TP+FP, B=TP+FN, C=TP+TN, D=TP+TN+FP+FN, E=FP+FN

V. Conclusion

In this paper, it was proved that the output of the Bisecting Min-Max DBSCAN algorithm was similar to that of the traditional DBSCAN algorithm thereby maintaining the quality of clustering and the time complexity was also reduced significantly. As a future scope the algorithm implemented by computing the values of epsilon and min points instead of just assigning them.

References

[1] W.J. Book, Modelling design and control of flexible manipulator arms: A tutorial review, *Proc. 29th IEEE Conf. on Decision and Control*, San Francisco, CA, 1990, 500-506.

[2] Terence Johnson, Santosh Kumar Singh, Divisive Hierarchical Bisecting Min–Max Clustering Algorithm, Advances in Intelligent Systems and Computing, Series Volume-468, Series ISSN 2194-5357, Online ISBN 978-981-10-1675-2, DOI 10.1007/978-981-10-1675-2_57 , 2016 International Conference on Data Engineering and Communication Technology -ICDECT 2016,March 10-11, LAVASA, Pune, Springer Singapore, copyright 2017, copyright holder Springer Science + Business Media Singapore, pp 579-592.

[3] Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu, A density-based algorithm for discovering clusters in large spatial databases with noise, ISBN 1-57735-004-9, Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD-96), AAAI Press, pp. 226–231.

[4] Terence Johnson, Santosh Kumar Singh, Quantitative Performance Analysis for the Family of Enhanced Strange Points Clustering Algorithms, International Journal of Applied Engineering Research, Series Volume 11, Series ISSN 0973-4562, Number 9 ,(2016), pp 6872-6880, Research India Publications.

[5] Martin Ester, Jörg Sander, Density-Based Clustering in Spatial Databases: The Algorithm GDBSCAN and Its Applications, Data Mining and Knowledge Discovery, Berlin, Springer-Verlag. 2 (2), pp. 169–194.

[6] R. J. G. B. Campello, D. Moulavi, J. Sander, Density-Based Clustering Based on Hierarchical Density Estimates, Proceedings of the 17th Pacific-Asia Conference on Knowledge Discovery in Databases, PAKDD 2013. Lecture Notes in Computer Science. 7819. p. 160. DOI:10.1007/978-3-642-37456-2_14. ISBN 978-3-642-37455-5.

Dr. Terence Johnson "The Bisecting Min Max DBSCAN Algorithm." IOSR Journal of Computer Engineering (IOSR-JCE) 20.3 (2018): 78-82.