Application of nearest neighbor graph in design and development of new density based clustering algorithm

Arvind Sharma[1] R K Gupta[2]

[1]Department of Information Technology
RJIT , Tekanpur

[2]Department of CSE
MITS, Gwalior

Abstract:

Spatial data Mining(SPDM) is a rising field in finding important and applicable trends for research and scientific purpose. Density based clustering is a part of spatial data mining. Density based clustering is being used by various researchers and agencies for identification of clusters in dense, non dense set of regions of data sets.we receive data in many form as images, tables, unstructured manner as blogs and mails etc. So, on the basis of concept of density based clustering algorithm , we can i. Handle and identify noise sample in spatial data sets, ii. To discover clusters with varying size and shape of data sets, and iii. To determine number of clusters for better performances in advance at the time of cluster generation. The Elbow method is very famous for this purpose and hence more suitable for unsupervised nature of algorithm. One of the most important characteristics of this research paper is to develop an algorithm that controls noise and generate clusters with arbitrary shapes and size while only DBSCAN (Density based Spatial clustering of applications with noise) shows reduced performance for the same database i.e. varying capacity of data sets. Therefore, in this research paper an advanced form of DBSCAN is proposed as NDBSCAN (New DBSCAN) which is more suitable for identifying clusters with varying densities. This algorithm is also based on adaptive nature of algorithm i.e. if a new pattern of cluster encounters or exists during the SPDM process then this algorithm will adapt it and store for future use.

Keywords: NDBSCAN, Eps, MinPts, SPDM, Clustering etc.

I. Introduction

In the fast-emerging technological scenario, the number and size of spatial databases related to geo-marketing, traffic control, weather forecasting, geographic information system, satellite imaging and environmental studies etc. are growing rapidly. The analysis of these databases for meaningful information or trends for decision making has generated an urgent requirement for new technologies & tools that can intelligently and efficiently process such databases to discover hidden, previously unknown and useful knowledge. Consequently, knowledge discovery techniques and tools related to spatial databases have become a critical area for research.

As mentioned earlier, Spatial data are being generated continuously by various sources and growing rapidly day by day. These data are stored and processed further with modern tools (Based on DM techniques) to discover hidden, previously unknown, and useful knowledge or patterns. This is a fundamental property of spatial data that each data has autocorrelations nearby its locations. So, different government agencies and research organizations are working on the analysis of such data with improved computational efficiency and acceptance of results for real-world applications. Hence, the basic need for spatial data analysis is the detection and prediction of hidden trends and patterns in a large volume of datasets. The prediction uses current database variables to forecast uncertain or potential consumer interest values. There are several DM techniques to complete this task efficiently. These techniques are:

Association, Classification, Sequential Patterns, Clustering, and Deviation detection etc.

In all spatial details, geometry is a key function. Geometry discusses the properties of an object in mathematics. These characteristics include measuring (metric), positions, lines, angles, surfaces, and solids (topology) and order. Simple geometry is traditionally developed from simpler geometry, such as points, lines, and regions.

- Relationships among spatial objects are found in an implicit manner, which requires deep drilling of data to identify meaningful trends.
- Availability of arbitrary shape and size which requires smart and highly learned algorithms.
- Performance acceptance is another issue in spatial data handling.
- Simulation is difficult for spatial data.

Now let us discuss in in other way, a density based clustering is the method identifying distinctive groups or clusters in a data set relied on the notion that a cluster is a dense contiguous region in the total data space, which is separate from other clusters by adjacent areas of relatively lower data density. The data points which are neither part of any clusters and nor the border points are called noise points.

In the very beginning DBSCAN was designed and implemented for this purpose. It is more effective and cited density based clustering algorithms which can determine clusters with noise and border points. It works with only Two(2) parameters i.e. Eps(Epsilon) and MinPts (minimum number of points) in a cluster. But still it is insensitive for huge size of dataset and varying capacities of data points. Outliers determination is also a challenge in DBSCAN.

It has noticed from different results of DBSCAN that results may vary unpredictable from for different values of Eps. As a result, the overall performance of DBSCAN degrades dramatically in the case of finding clusters with different data densities, because the Eps value , in DBSCAN is a global predetermined parameter. Therefore, it is more necessary to the whole process or system of SPDM that it should be able to identify different densities, arbitrary size and shapes in adaptive nature. Hence the idea of new algorithm as NDBSCAN takes place for future density based clustering algorithm which can overcome all deficiencies of DBSCAN.

II. Related work i.e. overview of DBSCAN performance

DBSCAN [66] Works as follows: The first arbitrary point p in database D is found with DBSCAN & in the given Eps all points close to point p. This algorithm provides a wrt, Eps & MinPts cluster if p is a key point. If p is a limiting point, the following object in the database must be accessed by DBSCAN. If two density clusters are closer to each other, merge two clusters. Where p is a central point, a new cluster will be created if the total no. of neighboring people is greater than minPts. This new cluster has been designated as point p and all its neighbors. The algorithm extracts the neighbors from the core areas of Eps later. All database points are accessed, and the process goes on. A complete overview of the algorithm indicates that the time complexity required by DBSCAN is O (n^2). For low dimensional space (kd-trees), this complexity can be reduced to O (nlog n), wherever n is no. of dots.[6][9][2].

The above procedure may be summarized as-

1. List all things as central points, boundaries, or points of noise
2. Eliminated noise points
3. Set the edge concerning each other amongst all the main points of Eps
4. Make a different cluster of linked core points for each group.

5. Allocate each border points to a category of the relevant group of clusters core points

DBSCAN offers major benefits [1]:

i. It can discover arbitrary-shaped clusters.
ii. There is no need to predict a cluster in advance, and thus it is more practical.
iii. It is difficult to use a greedy approach to substitute R*-tree data and greedy queries.
iv. Attributes are always selected and implemented to enhance time and spatial complexity.
v. The outliers are robust, and hybrid with other clusters is possible if they are like

After a thorough study of DBSCAN, major difficulties are identified as follows:

i. Cluster detection is tedious for various types of spatial data sets
ii. There are few approaches for evaluating the resemblance and no. of clusters as data sets for different purposes are used in advanced.
iii. Complexity to enhance and reduce inter-cluster distance.
iv. Problem with the identification of actual border and noise points.
v. If clusters with different densities are present, the findings are not consistent.
vi. If there are minor differences in the measurements of neighboring objects, then the major problem arises. The problem is of identifying adjacent clusters. The boundary object values can vary significantly from the opposite side of the same cluster.

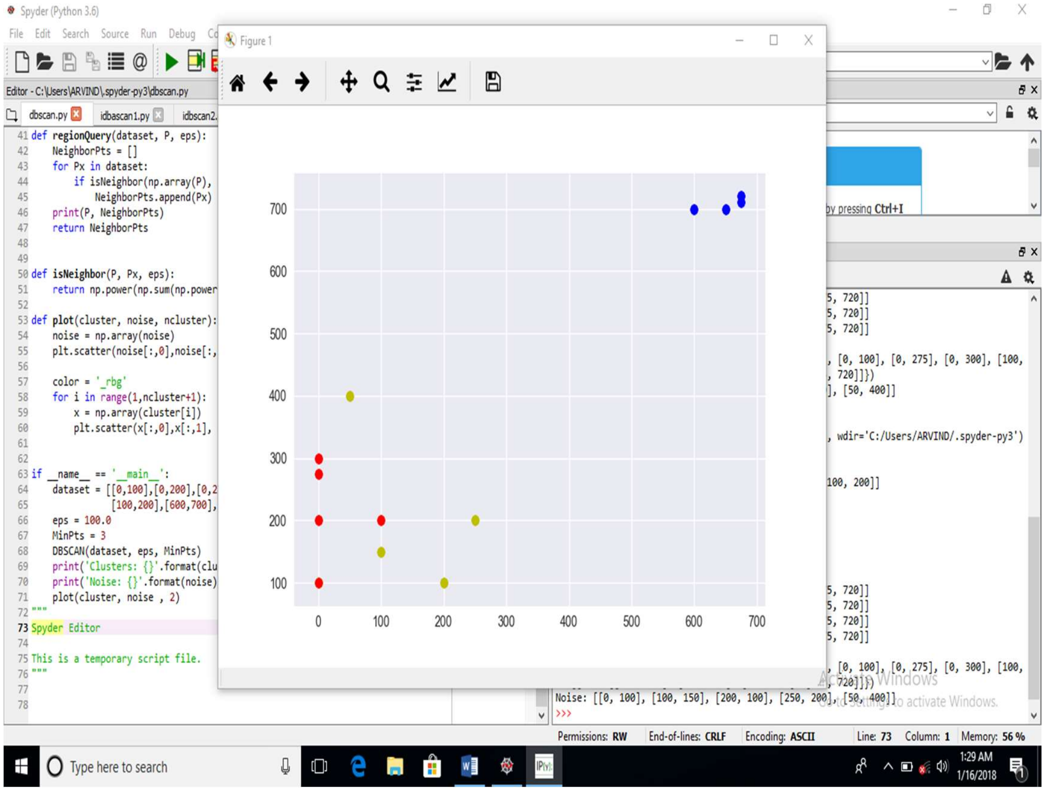
The author has implemented DBSCAN in python and its result is depicted with small synthetic data set for better understanding in the following current section.

A set of points (Table 1.1) are taken in the form of input and according to working program they form different clusters and noisy points according to their spatial neighborhood.

Table 1.1

```
[0, 100] [[0, 100], [0, 200]]

[0, 200] [[0, 100], [0, 200], [0, 275], [0, 300], [100, 200]]

[0, 275] [[0, 200], [0, 275], [0, 300]]

[0, 300] [[0, 200], [0, 275], [0, 300]]

[100, 200] [[0, 200], [100, 150], [100, 200]]

[100, 150] [[100, 150], [100, 200]]

[200, 100] [[200, 100]]

[250, 200] [[250, 200]]

[600, 700] [[600, 700], [650, 700], [675, 710], [675, 720]]

[650, 700] [[600, 700], [650, 700], [675, 710], [675, 720]]

[675, 710] [[600, 700], [650, 700], [675, 710], [675, 720]]

[675, 720] [[600, 700], [650, 700], [675, 710], [675, 720]]

[50, 400] [[50, 400]]

Clusters: defaultdict(<class 'list'>,

Core Point {1: [[0, 200], [0, 100], [0, 275], [0, 300], [100, 200]],

Noise : [[600, 700], [650, 700], [675, 710], [675, 720]]})

Border Point: [[600, 700], [100, 150], [200, 100], [250, 200], [50, 400]]>>>
```

In above paragraph the output of simple DBSCAN is shown by total 13 set of points as input and on the basis of working of Python program ,core points are shown by red color and border points are shown by yellow color while noise is shown by blue color



```
def regionQuery(dataset, P, eps):
    NeighborPts = []
    for Px in dataset:
        if isNeighbor(np.array(P),
           NeighborPts.append(Px)
    print(P, NeighborPts)
    return NeighborPts


def isNeighbor(P, Px, eps):
    return np.power(np.sum(np.power

def plot(cluster, noise, ncluster):
    noise = np.array(noise)
    plt.scatter(noise[:,0],noise[:,

    color = '_rbg'
    for i in range(1,ncluster+1):
        x = np.array(cluster[i])
        plt.scatter(x[:,0],x[:,1],


if __name__ == '__main__':
    dataset = [[0,100],[0,200],[0,2
                [100,200],[600,700],
    eps = 100.0
    MinPts = 3
    DBSCAN(dataset, eps, MinPts)
    print('Clusters: {}'.format(clu
    print('Noise: {}'.format(noise)
    plot(cluster, noise , 2)
"""
Spyder Editor

This is a temporary script file.
"""
```

```
5, 720]]
5, 720]]
5, 720]]

, [0, 100], [0, 275], [0, 300], [100,
, 720]]})
], [50, 400]]

, wdir='C:/Users/ARVIND/.spyder-py3')

100, 200]]

5, 720]]
5, 720]]
5, 720]]
5, 720]]

, [0, 100], [0, 275], [0, 300], [100,
, 720]]})
Noise: [[0, 100], [100, 150], [200, 100], [250, 200], [50, 400]]
>>>
```

Permissions: RW End-of-lines: CRLF Encoding: ASCII Line: 73 Column: 1 Memory: 56 %

Figure :A screen shot of output of DBSCAN algorithm

Basically the time complexity of DBSCAN algorithm is defined as O(n* time to find points in Eps-neighborhood). In worst case it may be O(n^2)[8]. To reduce time complexity of DBSCAN up to O(n logn) & also redesign it to support high dimensional data with varying densities[12]. It was main reason for selection of this algorithm for research work by the author.

III. Proposed new algorithm NDBSCAN for clustering

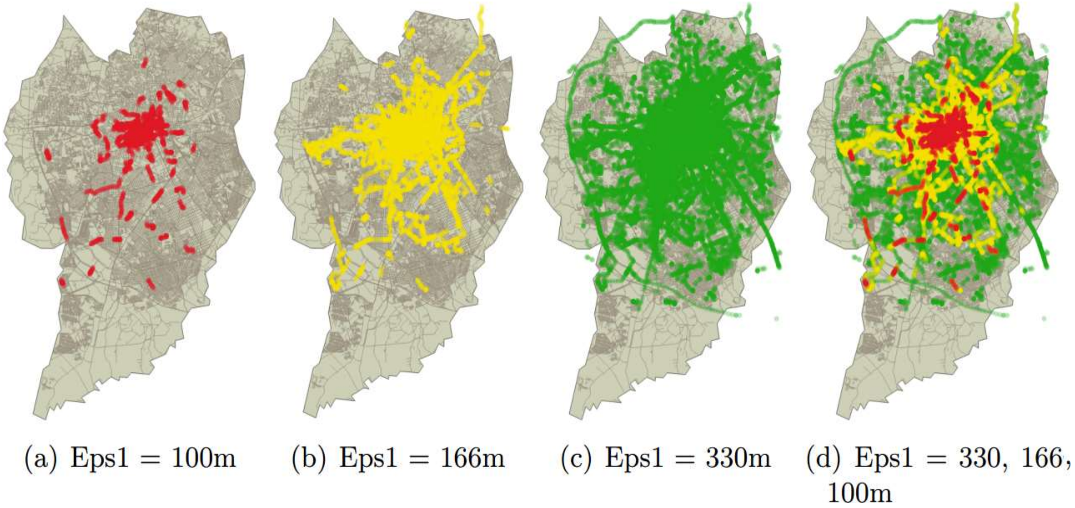
To overcome the deficiencies of DBSCAN algorithm we propose a new DBSCAN(NDBSCAN) algorithm that works well with varying data densities, and it can adapt the values of Eps and MinPts on the basis of the density distribution of the clusters. NDBSCAN can determine proper values of Eps and MinPts automatically. The value of Eps starts with a minimum random number and it increases in steps until we get proper distribution of points in the form of clusters. After that cluster has been saved separately and it is excluded from the main database. This process is repeated until we explore almost 100%(95%) data and make clusters accordingly and remaining data declared as noise points oroutliers. The proposed algorithm is given below

Algorithm:

```
 Input: Dataset D, Total  number of clusters and assume Eps , MinPts

   1. Select a point p from data base D
   2. Update Eps and Minpts from assumptions
   3. Loop
        C=1
      If recognized points>100%
      then make cluster and identify border points
      c=c++
      D=D-c
      While c<k
      Repeat the process
       End while
      Else
      Increase Eps and MinPts by.5
      Repeat
        D<= 5% Then
      Plot all the clusters
      Endif
      Endif

      End of Algorithm
```

IV. Result and discussion

Let us see input in the form of satellite image and apply different Eps values then system produces following results (Figure 1.1)-



(a) Eps1 = 100m (b) Eps1 = 166m (c) Eps1 = 330m (d) Eps1 = 330, 166, 100m

Figure 1.1: Pictorial form of clusters for different sea water properties with different Eps values.

In the above mentioned example marine environmental data(sea surface temperature,sea surface residual data, wave height values) collected from a variety of sources and then it has integrated as grids,shape files, and tables etc as shown in the model.Satellite images may be collected from NASA's site(http://podaac.jpl.nasa.gov/products/product109.html).

V. References

1. R H Güting.The international journal on very large database(VLDB),volume-3,issue-4,springer-verlag, pp 357-399.
2. Tie-li Yang,Ping Bai,Yu-Sheng Gong.Spatial data mining features between general data mining.IEEE,Shanghai,China.
3. Krzysztof Koperski; Junas Adhikary; and jiawai Han.Spatial data mining: progress and challenges,school of computer science simon fraser university Burnaby, B.C. Canada V5A IS6.
4. Arvind Sharma, R K Gupta.Intelligent knowledge discovery in spatial data sets .IJRECE Vol.4,issue 1,Jan-March. 2016.pp- 67-73.
5. M Ester,Hans Peter Kriegel,J sander,X Xu.Density connected sets and their applications for trend detection in spatial databases.KDD-97.
6. Jin Xingxing,cai Yingkun.Ma XiuJun at el.Novel method to integrate spatial data mining and geographic information system.IEEE 2005.pp. 764-767.

7. M may,and A. savinov.An integrated platform for spatial data mining and interactive visual analysis.Proceeding data mining 2002,Bologna,Italy.
8. Y xia,X X Fu.An improved approach and application for spatial data mining. IEEE ,2007. pp. 32-37.
9. Li D.R, wang S.L. at el. Theories and technologies of spatial data knowledge discovery.Geomatics and information science of Wuhan university,vol. 27, No.-3, pp. 221-233,2002.
10. Jain,Murty and Flynn.Geographic data mining and knowledge discovery.
11. Arvind Sharma, R K Gupta. Improved density based spatial clustering and applications with noise(IDBSCAN).Vol 2016,Article id 1564516.SCI Hindawi publication.
12. Arvind Sharma, R K Gupta. A survey of spatial data mining : Algorithms and architecture. pp. 15-22,2012.