Efficient Approaches for Density-Based Spatial Clustering of Applications with Noise

Pretom Kumar Saha, Doina Logofatu

Efficient Approaches for Density-Based Spatial Clustering of Applications with Noise

Pretom Kumar Saha and Doina Logofatu

Frankfurt University of Applied Sciences,
Nibelungenplatz 1,
D-60318 Frankfurt am Main, Germany.
saha@stud.fra-uas.de, logofatu@fb2.fra-uas.de

Abstract. A significant challenge for the growing world of data is to analyze, classify and manipulate spatial data. The challenge starts with the clustering process, which can be defined to characterize the spatial data with their relative properties in different groups or classes. This process can be performed using many different methods like grids, density, hierarchical and others. Among all these methods, the use of density for grouping leads to a lower noise data in result, which is called Density-Based Spatial Clustering of Applications with Noise (DBSCAN). The DBSCAN algorithm defines a data set in a group and separates the group from the other groups based on the density of the data surrounding the selection of data points. These data points and the density of the data are calculated depending on two parameters. One parameter is used as the radius of the data point to find the neighborhood data points. Another parameter is used to identify the noise in the collected data by keeping the minimum number of data points for the data density. Like other popular method k-means, DBSCAN does not require any input of the cluster number. It can sort the data set with the number of clusters according to data density. The purpose of this article is to explain the Efficient Density-based Spatial Clustering of Applications with Noise (DBSCAN) using a sample of data set, compare the results, identify the constraints, and suggest some possible solutions.

Keywords: Clustering · DBSCAN · Density-based Algorithms.

1 Introduction

An approach through which we can draw references from data sets consisting of input records without a specified classification is called the unsupervised learning method. Generally, this unsupervised learning method is used for creating a set of data in a meaningful structure, explanatory underlying processes, generative features, and groupings.

The technique for uniting related items in one group is clustering which is a crucial task in the field of data mining. Clustering can be utilized as an initial process for overall data mining or an independent technique. In the field of unsupervised learning methods, clustering can be used in several ways, e.g., outlier detection, data reduction, and identification of natural data types and classes.

Clustering is the endeavor to assemble info in a hasty manner that meets with human instinct. Tragically, our self-generated thoughts of what makes a 'cluster' are inefficaciously characterized and passing settings delicate [1]. This comes concerning in an exceedingly lots of cluster calculations each of that match a somewhat distinctive natural plan of what a typical gathering is [2]. In spite of the vulnerability basic, the cluster prepares its takings to be used in a very immense range of logical areas. The fundamental issue of finding groupings is inevitable and comes concerning, in any case impoverished, are still imperative and enlightening [2]. It's used in several areas like atomic flow [14], plane flight manner investigation [15], natural philosophy [16], and social analytics [17], among varied others. Whereas cluster has varied activities to different people, our specific point is on clump for the explanation of investigating data analysis [2].

In the field of Data analysis, numerous conventional clustering calculations are ineffectively suited. In specific, most clustering calculations endure from the issues of troublesome parameter determination, inadequately strength to commotion within the information, and distributional presumptions approximately the clusters themselves [2]. Numerous calculations require the choice of the number of clusters, either expressly, or implicitly through intermediary parameters. Within the larger part of utilizing cases we have experienced, selecting the number of clusters is exceptionally troublesome a priori [2]. Strategies to decide the number of clusters such as the elbow strategy and outline strategy are frequently subjective and can be difficult to apply in the hone. Eventually, these strategies all pivot on the clustering quality degree chosen; these are different and frequently profoundly related with specific clustering calculations [1].

There are six techniques that can be actualized for clustering namely partitioning, hierarchical, density, grid, model, and constraint-based models [9]. Since the density-based strategy bunch data objects based on comparative density locale, it is exceptionally viable and more reasonable for spatial databases [9]. It considers a cluster as a high-density region when compared to its enveloping locale. In huge spatial database applications, the clustering calculations require to take following necessities [18]:

1. Less space information to choose input parameters
2. Recognizing subjective formed clusters
3. Great adequacy on huge databases [9].

Depending on six vital variables such as time complexity, input parameters, taking care of shifted thickness, dealing with of self-assertive shape, vigor to clamor, and insensitiveness to information input arrange, there exist fifteen algorithms like DBSCAN, DBCLASD, GDBSCAN, DENCLUE, OPTICS, DBRS, IDBSCAN, VDBSCAN, LDBSCAN, ST-DBSCAN, DDSC, DVBSCAN, DBSC, DMDBSCAN, and DCURS [9].

2 General Description of DBSCAN Algorithm

Density Based Spatial Clustering of Application with Noise (DBSCAN) is a concept that uses density to cluster data [17]. That means the minimum density of data depending on a certain feature is used as a cluster. The main advantage of this method is that it can identify clusters without entering the required cluster number. DBSCAN algorithm depend on to variable the radius (eps) and the minimum number of data nodes in a certain place (min_{pts}) [3]. The following conditions are used concerning these two variables:

Neighborhood points: All the nodes, which presents in the range of radios *eps* neighborhood from center node p is the definition as neighborhood node of p. These neighborhood nodes are classified into core node and border node.

Core node $=neighbour_{pts} > min_{pts}$.
Border Node $= neighbour_{pts} < min_{pts}$.

Here, $neighbour_{pts}$ is the number of neighborhood node respect of center node p [2].

Direct density reachable: If there is a connection between two nodes p and q, through which direct travel is possible from p to q. if p=p1 and q= pn then the direct travel path p1, p2, p3. . . . . . and pn. Then, this p and q node are called direct density reachable [4].

Density connected: If two-node p and q are individually density reachable from a common node o within the same min_{pts} and eps value, then this p and q node are density connected [3].

Cluster: Node p and q are in the same cluster C of a dataset define by min_{pts} and eps. if $p\epsilon C$ and q is density connected or density reachable from p then $q\epsilon C$ [13].

Noise: If p is a node of the dataset but does not belong in any cluster of that dataset concerning min_{pts} and eps, Then, this p node is called Noise [13].

3 DBSCAN Algorithm Details

K-means is one of the foremost well-known clustering algorithms. However, the biggest problem with K-means clustering is that it cannot determine the number of clusters. Based on this problem, DBSCAN has the advantage that it can identify the number of clusters in the data set.

Algorithm 1: Configuring Parameters for DBSCAN

```
Result: Return A dictionary with $eps, min_{pts}, dim$
initialization read config file ;
for $line$ to $lines$ do
    if $line$ $! = \#$ then
        $eps, min_{pts}, dim \leftarrow line$;
    else
        continue;
    end
end
```

The proposed clustering is implemented in PYTHON on a 2-dimensional data set. The implementation starts with the configuration of the DBSCAN algorithm, which includes the maximum allowable distance for neighboring points (*eps*), the minimum number of neighbors surrounding a point (min_{pts}), and the dimension of the data set [8] in Algorithm 1.

The next step is processing the data set and refine them so that the algorithm can be established and find out the cluster from the data set in Algorithm 2.

Algorithm 2: Check Data Set

```
Result: Return A List of $data$ in dimension $dim$
initialization read csv data file ;
initialization number of data $rows$ ;
for $row$ to $rows$ do
    if $isinstance(row, str)$ then
        continue;
    else
        $data \leftarrow row$ ;
    end
end
```

The algorithm shows in the Algorithm **??** bellow is the DBSCAN written in Python. At the begging of the algorithm, it selects one fresh point randomly that is not used to calculate the density of data yet depending surrounding that point in the radius of *eps* neighborhood. By finding out the actual center point of the density, DBSCAN defines the center point of the density with a cluster number and also defines the other direct reachable points in that density with a similar cluster number. The points will be defined as noise if it is not directly reachable from the core point. The whole process will be continuing until all the points of the data set are not defined with at least one cluster. All the points do not belong in any cluster that points will be classified as noise. The DBSCAN iterate all the points of the data set, so if the data set has n number of data points then the computational complexity of the whole algorithm is $O(n)^2$ [10] .

Algorithm 3: DBSCAN

```
Result: Plot Data Clusters with different Colors
initialization $data,params$;
initialization $noise$;
for $point$ to $data$ do
    if $not\ visited$ then
        $point \leftarrow visited$;
        $neighbour_{pts} \leftarrow$ distance between two points in $eps$;
        if $len(neighbour_{pts}) < min_{pts}$ then
            $noise \leftarrow point$;
        else
            $cluster \leftarrow point$;
        end
    end
end
```

4 Performance and Evaluation

Source of data collection is https://www.kaggle.com/ and for this project I select a numeric data set with 2 dimensions. The information is collected within the shape of .csv. The flowchart of DBSCAN and execution results are appeared in the Figures 1, 2, 3 and 4 .

At the beginning, 1000 data is applied on this DBSCAN and the figure 2 represents the output for this 1000 data.

Figure 3 represents the output of 10000 data. Different colors are used here to make it easier to the identification of noise and clusters. The blue color represents the noise and other colors for clusters.

If we try to explain the result in mathematical way then it can say that eps with 10 and min_{pts} with 5 the DBSCAN gives 143 clusters. Where the total data points are 10000 including 634 noise points.

First, The algorithm was overviewed as a flowchart figure 1, then with some sequence of pseudocode 3 and finally with python scripts [15]. Here one example is presented to understand the process of DBSCAN with input and output forms:

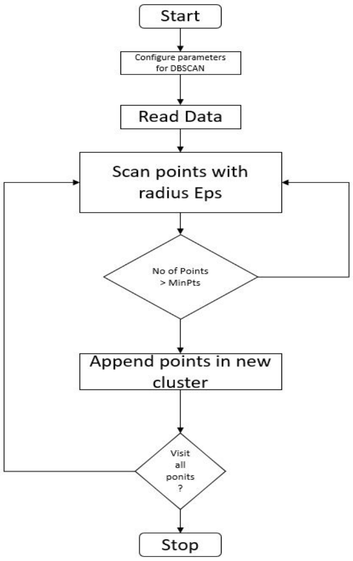

Start
Configure parameters for DBSCAN
Read Data
Scan points with radius Eps
No of Points > MinPts
Append points in new cluster
Visit all ponits ?
Stop

Fig. 1. Flowchart of the DBSCAN Algorithm

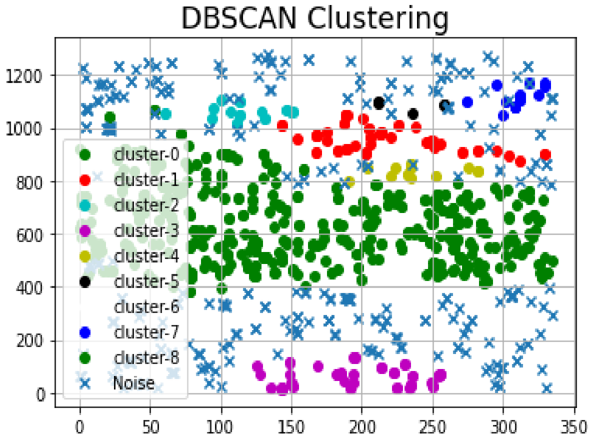


Fig. 2. DBSCAN Cluster using 1000 data

From the above discussion, for the actual data it does not looks the same but anyhow the overall procedure of density-based concept is the same. DBSCAN is appropriate for any numeric data. It has been taken note that, for huge information measure, the era of comes about corrupted within the shape logarithmic nature. Two screenshots are given here which represent the output of DBSCAN using the actual data set. First One is for the 1000 data figure 2and the 2nd is

```
$(0,10)(0,20)(0,27)(10,15)(20,10)(25,20)(0,30),$
  $(10,20)(60,70)(65,70)(67,71)(67,72),(5,40).$;
$eps = 10.0$;
$min_{pts} = 3$;
Output Cluster points;
$Cluster1 - (0,10)(0,20)(0,27)(10,15)(0,30)(10,20)$;
$Cluster2 - (60,70)(65,70)(67,70)(67,71)(67,72)$;
Noise points $-(20,10)(25,20)(5,10)$;
```

for the 10000 figure 3.

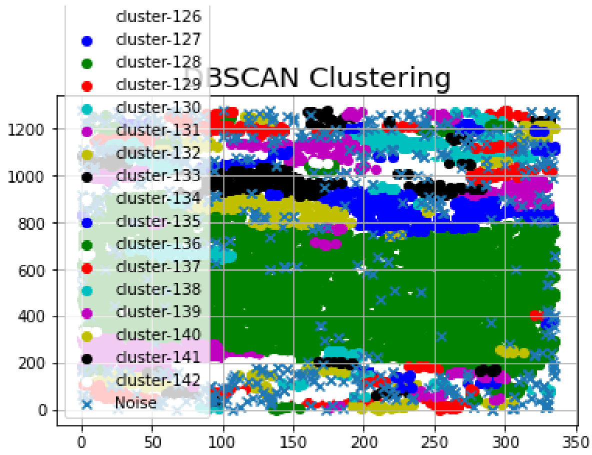


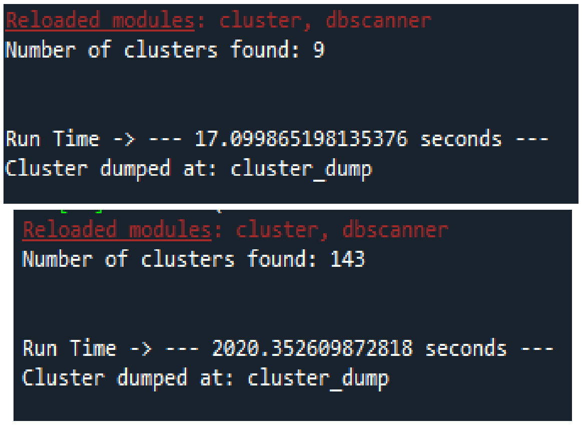
Fig. 3. DBSCAN Cluster using 10000 data

As shown in Figure 4 the DBSCAN is better in terms of small numbers of data and run time is also small. DBSCAN makes a direct increment in computing time related to the number of data in the data set. Figure 4 is produced on different measure of information units and comes about are computed at common computer machine. Confinement and future perspectives are said within the another stage.

DBSCAN may be summarized as follows:

- DBSCAN is more exact in the density reachability approach.
- DBSCAN can discover self-assertive formed clusters. It gives way better execution for near focuses having different properties and exceptionally near clusters at the boundary, that's, border points.
- Do not require the number of clusters (k) in advance.

- DBSCAN requires 2 parameters.
- Outlier discovery is much superior in DBSCAN.



```
Reloaded modules: cluster, dbscanner
Number of clusters found: 9


Run Time -> --- 17.099865198135376 seconds ---
Cluster dumped at: cluster_dump
```

```
Reloaded modules: cluster, dbscanner
Number of clusters found: 143


Run Time -> --- 2020.352609872818 seconds ---
Cluster dumped at: cluster_dump
```

Fig. 4. Run time representation for data 1000 and 10000

5 Drawbacks of DBSCAN

We are at the center of discussion on the DBSCAN, which is proposed by Ester [18]. DBSCAN is one of the foremost broadly utilized calculations in numerous applications, such as chemistry, spectroscopy, social science, gracious designing, peculiarity location, therapeutic and biomedical picture examination [13]. As a pioneer of density-based clustering calculations, it has the same nonignorable confinements as the conventional density-based calculations which have been specified here.

- The execution of clustering depends on two indicated parameters. It is troublesome to assess fitting values of these two parameters for different datasets without any sufficient earlier information [13].
- The computational complexity is high when managing with a high dimensional dataset [11].
- This algorithm prefers proper arranging dataset, distinctive orderings of data in the same dataset distort the result [6].
- Due to the use of global density parameter, adjoining clusters of diverse densities cannot be appropriately distinguished [13].

6 Analogous Evolution Of DBSCAN

In this segment, the related development of density-based algorithms is introduced. Here the focus on the DBSCAN calculation beneath the density-based clustering calculations [18]. To facilitate understanding of the content of this article, some approaches already proposed to extend and improve the DBSCAN calculation are examined in detail here. The density around a protest is accomplished by checking the number of objects in a locale of indicated parameter radius, say *eps*, around the object [18]. An object is treated as dense if it is having *eps* neighborhood of that object more noteworthy than or break even with to an indicated edge least objects (u), something else scanty (non-core). Non-core objects that indicated radius are known as DBSCAN may have wide variety [18]. Such clusters may be spoken to by a few smaller clusters so that each cluster may have a sensibly uniform density. DBSCAN does not characterize the upper constrain of a center object i.e. how much objects may display in $neighbour_{pts}$. So due to this if there's a wide variety in local thickness. it'll consolidate into the same have a center protest inside the noise.

OPTICS [19]: In full form Ordering Points To Identify the Clustering Structure algorithm for the reason of cluster examination which does not deliver a clustering of a data set unequivocally, but instep makes an increased ordering of the database speaking to density-based clustering structure. This cluster-ordering contains data that is proportionate to the density-based clustering comparing to a wide run of parameter settings. Be that as it may, OPTICS needs another algorithm beside it to deliver unequivocal clusters.

DENCLUE [20]: This algorithm follows the process of DBSCAN with some improvements. It clusters the dataset independently with Eps and Minpts, then blends joined and comparative clusters together agreeing to the given edge. The other difference is that this algorithm uses a very efficient grid method. The main difficulty is that it depends on a large number of input parameters.

VDBSCAN [21]: The essential approach of this algorithm is the use of k-dist to decide the parameters Eps and MinPts is to see at the behavior of the separate from a point to its kth closest neighbor. The k-dists are computed for all the information focuses for a few k, sorted in climbing arrange, and after that plotted to utilize the sorted values, as a result, a sharp alter is anticipated to see. The sharp alter at the esteem of k-dist compares to the appropriate esteem of Eps.

DD DBSCAN [22]: One of the updated forms of DBSCAN which apply the upper restrain amid the development of a cluster. The clusters are created in distinctive shapes, sizes and vary in neighborhood density [22].

But this calculation is having a downside that still it cannot handle the density variety inside the cluster. On the off chance that we watch the cluster arrangement at that point, it too implies the wide density variety inside the cluster.

WaveCluster [10]: This Density base algorithm mostly works on low dimensional data. It's one kind of grid-based algorithm and applies wavelet transform to the feature space. To find clusters in different scales of shape the required complexity is O(n) for the WaveCluster algorithm.

CLIQUE [11]: The CLIQUE algorithm is used the partitional or hierarchical clustering techniques. It can be implemented for distance-based or connectivity-based. Some special improvements are developed for the case of high-dimensional data such as Irrelevance of distances, Sparsity of the data, and different features or a different correlation of features that may be relevant for varying clusters.

DDSC [23]: Once more an expansion of the DBSCAN to distinguish clusters of diverse shapes, sizes, and vary in neighborhood density. Clusters recognized by it are having non-overlapped spatial district regions with sensible homogeneous thickness varieties inside them. Adjoining locales are isolated into diverse clusters in case there's a critical alter in densities. The clusters may be touching i.e. not isolated by any meager locale as required by DBSCAN. In this way, characteristic clusters in a dataset can be extricated. An included advantage is that the affectability of the input parameter X which is a vital impediment of DBSCAN is diminished essentially.

CHAMELEON [24]: Algorithm finds clusters of datasets by the two-phase algorithm. Firstly, it produces a k-nearest neighbor graph. Finally, it blends comparative sub-clusters.

DENCLUE [6]: Another upgrade of the DBSCAN calculation is DENCLUE [5], it calculates an impact work that portrays the effect of a question upon its neighborhood. The calculation presents the scientific establishment and scales well since it can prepare exceedingly inadequate datasets with the slightest work.

CURD [10]: CURD captures the shape and degree of a cluster by references, and after that analyzes the data based on the references. It can discover clusters with subjective shapes and is uncaring to noise information. Mining exceptionally huge databases are its goal; however, the adequacy could be an issue.

ST-DBSCAN [10]: ST-DBSCAN is an expansion of DBSCAN to handle spatial-temporal datasets. It reclassifies border points to find adjoining clusters and commotion centers among abutting clusters. ST-DBSCAN does not handle moved densities well.

There are many other density-based algorithms, all developed to solve the limitations of DBSCAN. However, each algorithm also has some limitations. In this paper, an overview of DBSCAN is given with the differences between the other updated algorithms.

7 Conclusion

In this literature, we presented a density-based clustering algorithm. The following issues are unraveled by DBSCAN:

- Does not require a-priori determination of number of clusters
- Able to recognize noise information whereas clustering
- DBSCAN calculation is able to discover subjectively measure and subjectively molded clusters [12].

DBSCAN is proposed to handle specified clustering issues. Other algorithms attempted to fathom these issues sometime recently like DBSCAN, but still have concerns with different-densities databases and adjoining clusters as well because it includes unused parameters other *eps* and min_{pts}. DBSCAN works with two parameters only and demonstrated its capacity to overcome clustering issues-particularly densities clusters. There are a few openings for future inquire. For illustration on the off chance that DBSCAN is begun with distinctive center focuses, there will be a few borders focuses that will be created into diverse clusters. Presently, these border focuses areas were relegated to the cluster found to begin with. These border focuses don't fundamentally have a place in the certain cluster and they can be allotted to a few clusters at the same time. In expansion, the current satisfactory density extend is calculated agreeing to the density of the current point. It might abdicate a more sensible result on the off chance that weights are allotted to the focuses as that allotted to the same cluster of the current point.

References

1. C. Hennig.: What are the true clusters? In: Pattern Recognition Letters, vol. 64, pp. 53 – 62, 2015, philosophical Aspects of Pattern Recognition. [Online]. Available: http://www.sciencedirect.com/science/proc/pii/S0167865515001269
2. Leland McInnes and John Healy.:Accelerated Hierarchical Density Based Clustering. In: International Conference on Data Mining Workshops, IEEE, 2017.
3. Wei-Tung Wang, Yi-Leh Wu, Cheng-Yuan Tang and Maw-Kae Hor.:Adaptive density-based spatial clustering of applications with noise (DBSCAN) according to data. In: Conference on Machine Learning and Cybernetics (ICMLC), IEEE, 2015.

4. Md Farhadur Rahmanz, Weimo Liuy, Saad Bin Suhaimy, Nan Zhangy, Saravanan Thirumuruganathanz and Gautam Das.: HDBSCAN: Density based Clustering over Location Based Service. In: arXiv:1602.03730v2, 2016.
5. Mohammad F. Hassanin, Mohamed Hassan and Abdalla Shoeb.: DDBSCAN: Different Densities-Based Spatial Clustering of Applications with Noise. In: International Conference on Control,Instrumentation, Communication and Computational Technologies (lCCICCT), IEEE, 2015.
6. Ozge Uncu, William A. Gruver, Dilip B. Kotak, Dorian Sabaz, Zafeer Alibhai and Colin Ng.: GRIDBSCAN: GRId Density-Based Spatial Clustering of Applications with Noise. In: International Conference on Systems, Man, and Cybernetics October 8-11, 2006, Taipei, Taiwan, IEEE, 2016.
7. Anant Ram, Ashish Sharma, Anand S. Jalall, Raghuraj Singh and Ankur Agrawal.: An Enhanced Density Based Spatial Clustering of Applications with Noise. In International Advance Computing Conference (IACC 2009) Patiala, India, IEEE, 2016.
8. Satyasai Jagannath Nanda and Ganapati Panda.: Design of Computationally Efficient Density-based Clustering Algorithms. In: Data and Knowledge Engineering, 2014.
9. K. Nafees Ahmed and T. Abdul Razak.: A Comparative Study of Different Density based Spatial Clustering Algorithms. In: International Journal of Computer Applications, Volume 99– No.8, August 2014.
10. Lian Duan, Lida Xu, Feng Guo, Jun Lee and Baopin Yan.: A local-density based spatial clustering algorithm with noise. In: L. Duan et al. / Information Systems 32, Elsevier B.V, 2006.
11. Peng Liu, Dong Zhou and Naijun Wu.: VDBSCAN: Varied Density Based Spatial Clustering of Applications with Noise. In: School of Information Management and Engineering, Shanghai University of Finance and Economics, Shanghai, 200433, China, IEEE, 2007.
12. Arvind Sharma, R. K. Gupta, and Akhilesh Tiwari.: Improved Density Based Spatial Clustering of Applications of Noise Clustering Algorithm for Knowledge Discovery in Spatial Data. In: Hindawi Publishing Corporation Mathematical Problems in Engineering, Volume 2016, proc ID 1564516, 9 pages.
13. Yinghua Lv, Tinghuai Ma, Meili Tang , Jie Cao, Yuan Tian, Abdullah Al-Dhelaan and Mznah Al-Rodhaan.: An efficient and scalable density-based clustering algorithm for datasets with complex structures. In: Y. Lv et al. / Neurocomputing, Elsevier B.V, 2015.
14. R. L. Melvin, R. C. Godwin, J. Xiao, W. G. Thompson, K. S. Berenhaut,and F. R. Salsbury Jr.: Uncovering large-scale conformational change in molecular dynamics without prior knowledge. In: Journal of Chemical Theory and Computation, vol. 12, no. 12, pp. 6130–6146, 2016.
15. A. T. Wilson, M. D. Rintoul, and C. G. Valicka.: Exploratory trajectory clustering with distance geometry. In: International Conference on Augmented Cognition. Springer, 2016, pp. 263–274.
16. P. R. Spackman, S. P. Thomas, and D. Jayatilaka.: High throughput profiling of molecular shapes in crystals. In: Scientific reports, vol. 6, 2016.
17. M. Korakakis, P. Mylonas, and E. Spyrou.: Xenia: A context aware tour recommendation system based on social network metadata information. In: Semantic and Social Media Adaptation and Personalization (SMAP), 2016 11th International Workshop on. IEEE, 2016, pp. 59–64.
18. 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 of 2nd Int. Conf. on Knowledge Discovery and Data Mining (KDD-96), 1996.

19. M. Ankerst, M. Breunig, H. P. Kriegel, J. Sander.: OPTICS: Ordering Objects to Identify the Clustering Structure. In: Proc. ACM SIGMOD. In International Conference on Management of Data, pp. 49-60. 1999.
20. A. Hinneburg and D. Keim.: DENCLUE: An efficient approach to clustering in large multimedia data sets with noise.In 4th International Conference on Knowledge Discovery and Data Mdining, pp. 58-65. 1998.
21. Peng Liu, Dong Zhou, Naijun Wu.: VDBSCAN:Varied Density Based Spatial Clustering of Applications with Noise. In: Proceedings of IEEE ICSSSM2007, pp.528-531, 2007.
22. B.Borach, D.K. Bhattacharya.: A Clustering Technique using Density Difference. In: ICSCN. India Feb(22-24) pp-585-588, 2007.
23. B. Borah, D.K. Bhattacharyya.: DDSC, "A Density Differentiated Spatial Clustering Technique".In: Journal Of Computers, Vol. 3, No. 2, February 2008.
24. G.Karypis, E.H. Han, V. Kumar.: CHAMELEON: A Hhierarchical Clustering Algorithm using Dynamic Modeling. In: Computer 32(8): 68-75, 1999.