Journal of Software Engineering Research
Jan-June 2011,
www.journalshub.com

FDBSCAN

[#] D.N.V.Kranthi kumar
*M.C.A Lecturer,*K.B.N.College, vijayawada
kranthihcu@gmail.com

V.S.K.Chandra sekhar
*M.C.A Lecturer*K.B.N.College, vijayawada.
Krishna_kris007@rediffmail.com
[#] SK.Johnbee., M.Sc (computers)
*Lecturer*K.B.N.College, vijayawada

Abstract: **The aim of FDBSCAN is to generate clusters based on density. The prerequisite for generating clusters is to convert data into a normalized form, so that all attributes become unit less variables. To normalize the data, data must be in a consistent form. If not, some data pre-processing techniques have to be used. In this project two pre-processing techniques are used, namely "Filling Missing Values" and "Attribute Construction" before proceeding to normalization.After the Data Cleaning is performed, normalization is done on the data. In this project, z-score normalization technique is used. Dissimilarities between attributes is calculated and stored in a two dimensional array, known as distance matrix. In the end, FDBSCAN algorithm is applied based on the distance matrix. The algorithm takes two parameters, Eps-neighborhood and Minimum number of points. Based on these two parameters, clusters are generated for the given data set.**

Key Words: pre-processing, Z-score, distance matrix

I. INTRODUCTION

DBSCAN (Density-Based Spatial Clustering of Applications with Noise) is a data clustering algorithm proposed by Martin Ester, Hans-Peter Kriegel, Jörg Sander and Xiaowei Xu in 1996. It is a density-based clustering algorithm because it finds a number of clusters starting from the estimated density distribution of corresponding nodes. DBSCAN is one of the most common clustering algorithms and also most cited in scientific literature.

Basic IDEA

DBSCAN's definition of a cluster is based on the notion of density reachability. Basically, a point q is directly density-reachable from a point p if it is not farther away than a given distance ε (i.e., is part of its ε-neighborhood), and if p is surrounded by sufficiently many points such that one may consider p and q be part of a cluster. q is called density-reachable from p if there is a sequence $p_1, \ldots, p_n$ of points with and p1 = p and pn = q where each pi + 1 is directly density-reachable from pi. Note that the relation of density-reachable is not symmetric (since q might lie on the edge of a cluster, having insufficiently many neighbors to count as a genuine cluster element), so the notion of density-connected is introduced: two points p and q are density-connected if there is a point o such that o and p as well as o and q are density-reachable.

A cluster, which is a subset of the points of the database, satisfies two properties:

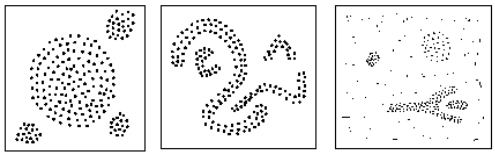
All points within the cluster are mutually density-connected.

If a point is density-connected to any point of the cluster, it is part of the cluster as well.

II CLUSTERING ALGORITHM

Partitioning Alg: Construct various partitions then evaluate them by some criterion (CLARANS, O(n) calls)
Hierarchy Alg: Create a hierarchical decomposition of the set of data (or objects) using some criterion (merge & divisive, difficult to find termination condition)
Density-based Alg: based on local connectivity and density functions

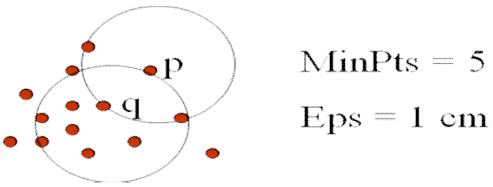
III DENSITY-BASED CLUSTERING



Algorithm Description

DBSCAN requires two parameters: ε (eps) and the minimum number of points required to form a cluster (minPts). It starts with an arbitrary starting point that has not been visited. This point's ε-neighborhood is retrieved, and if it contains sufficiently many points, a cluster is started. Otherwise, the point is labeled as noise. Note that this point might later be found in a sufficiently sized ε-environment of a different point and hence be made part of a cluster.

If a point is found to be part of a cluster, its ε-neighborhood is also part of that cluster. Hence, all points that are found within the ε-neighborhood are added, as is their own ε-neighborhood. This process continues until the cluster is completely found. Then, a new unvisited point is retrieved and processed, leading to the discovery of a further cluster or noise.



Major features:

- Discover clusters of arbitrary shape
- Handle noise
- One scan

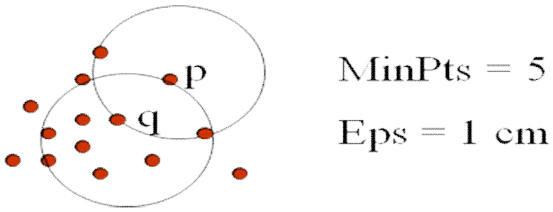
Density Concepts

Two global parameters***:***

- ***Eps***: Maximum radius of the neighbourhood
- ***MinPts***: Minimum number of points in an Eps-neighbourhood 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



Density-Based Clustering: Background

$N_{Eps}(p)$: ***{q belongs to D | dist(p,q) <= Eps}***

Directly density-reachable**:** A point ***p*** is directly density-reachable from a point ***q*** wrt. ***Eps***, ***MinPts*** if

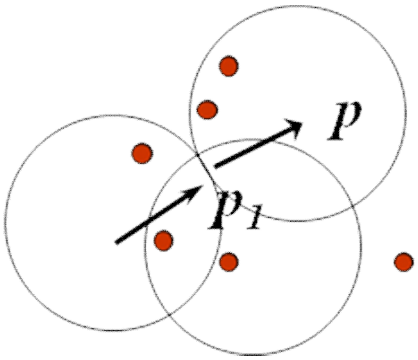
- 1) ***p*** belongs to ***$N_{Eps}(q)$***
- 2) ***$|N_{Eps}(q)|$*** >= ***MinPts***

(core point condition)

Density-Based Clustering: Background (II)

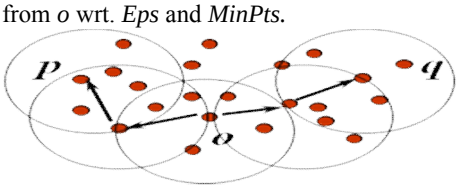
Density-reachable:

A point *p* is density-reachable from a point *q* wrt. *Eps*, *MinPts* if there is a chain of points $p_1, \ldots, p_n$, $p_1 = q$, $p_n = p$ such that p_{i+1} is directly density-reachable from p_i



Density-connected

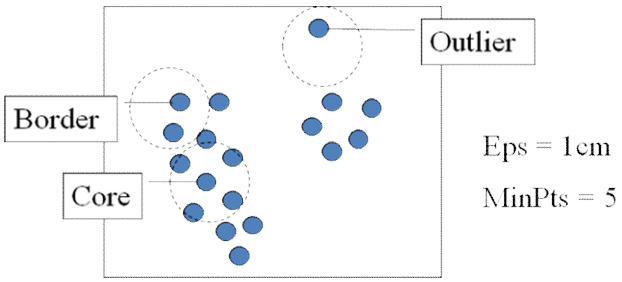
A point *p* is density-connected to a point *q* wrt. *Eps*, *MinPts* if there is a point *o* such that both, *p* and *q* are density-reachable from *o* wrt. *Eps* and *MinPts.*



DBSCAN: Density Based Spatial Clustering of Applications with Noise

Relies on a *density-based* notion of cluster: A *cluster* is defined as a maximal set of density-connected points

Discovers clusters of arbitrary shape in spatial databases with noise



IV DISADVANTAGES OF DBSCAN

DBSCAN can only result in a good clustering as good as its distance measure is in the function getNeighbors(P,epsilon). The most common distance metric used is the euclidean distance measure. Especially for high-dimensional data, this distance metric can be rendered almost useless due to the so called "Curse of dimensionality", rendering it hard to find an appropriate value for epsilon. This effect however is present also in any other algorithm based on the euclidean distance.

DBSCAN cannot cluster data sets well with large differences in densities, since the minPts-epsilon combination cannot be chosen appropriately for all clusters then.

DBSCAN: The Algorithm (1)

- Arbitrary select a point p
- Retrieve all points density-reachable from p wrt 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.

DBSCAN: The Algorithm (2)

```
DBSCAN (SetOfPoints, Eps, MinPts)

// SetOfPoints is UNCLASSIFIED
  ClusterId := nextId(NOISE);
  FOR i FROM 1 TO SetOfPoints.size DO
    Point := SetOfPoints.get(i);
    IF Point.ClId = UNCLASSIFIED THEN
      IF ExpandCluster(SetOfPoints, Point,
            ClusterId, Eps, MinPts) THEN
        ClusterId := nextId(ClusterId)
      END IF
    END IF
  END FOR
END; // DBSCAN
```

DBSCAN: The Algorithm (3)

```
ExpandCluster(SetOfPoints, Point, ClId, Eps,
              MinPts) : Boolean;
  seeds:=SetOfPoints.regionQuery(Point,Eps);
  IF seeds.size<MinPts THEN // no core point
    SetOfPoint.changeClId(Point,NOISE);
    RETURN False;
  ELSE   // all points in seeds are density-
         // reachable from Point
    SetOfPoints.changeClIds(seeds,ClId);
    seeds.delete(Point);
    WHILE seeds <> Empty DO
```

Algorithm:4

```
      currentP := seeds.first();
      result := SetOfPoints.regionQuery(currentP,
                                        Eps);
      IF result.size >= MinPts THEN
        FOR i FROM 1 TO result.size DO
          resultP := result.get(i);
          IF resultP.ClId
              IN {UNCLASSIFIED, NOISE} THEN
            IF resultP.ClId = UNCLASSIFIED THEN
              seeds.append(resultP);
            END IF;
            SetOfPoints.changeClId(resultP,ClId);
          END IF; // UNCLASSIFIED or NOISE
        END FOR;
      END IF; // result.size >= MinPts
      seeds.delete(currentP);
    END WHILE; // seeds <> Empty
    RETURN True;
  END IF
END; // ExpandCluster
```

V FDBSCAN

A FAST DBSCAN ALGORITHM

FDBSCAN is a fast version of the original DBSCAN algorithm. In DBSCAN, when the first core point is found in a new cluster, the first batch of representative points is selected as seed points for cluster expansion and in the subsequent iteratons, more representative seeds are added for cluster expansion till no more representative seed can be found, which means the cluster expansion is finished. Following is the outline of FDBSCAN algorithm.

```
FDBSCAN(SetPoints,Eps,Minpts,representative-minpts)
{
  ClusterId:=nextId(noise);

  For i:=1 to SetofPoints.size
```

```
Do
  {
   Point:=SetofPoints.get(i)
    If Point:=C1Id=UNCLASSIFIED

      Then

        If ExpandCluster (SetofPoints, Point, ClusterId, Eps, MinPts, Representative-minpts)

  Then

 ClusterId:=nextId(ClusterId)

 }
 }
```

CONCLUSION

Clustering, in data mining, is a useful techniqu for discovering intrerestin data distributions and patterns in the underlying data. As an outstanding representative of clustering algorithms, DBSCAN algorithm shows good performance in clustering spatial data. Based on the orginal DBSCAN algorithm. By selecting only a small number of representative points in a core points neighborhood as seeds to expand cluster, FDBSCAN executes less region queries than DBSCAN does, which reduces clustering time and I/O cost. We performed a performance evaluation on synthetic data and real data of the SEQUOTA 2000 benchmark. The experimental results show that FDBSCAN is faster than the original DBSCAN algorithm by several times.

Future research will have to consider the following issues. Firstly, extend the FDBSCAN to high-dimensional data space. Secondly, integrate data sampling, data partitioning and parallel techniques with DBSCAN or FDBSCAN to cluster very large scale databases. Thirdly, establish an adaptive and interactive density-based clustering algorithm, which does not need the user to input any heuristic parameter.

REFERENCES

[1] JkWei **Hm,** Micheline Kamber. Data Mining Concepts **and Techniques.** China Machine **Press.** 2004,pp242-245

[2] G.H. Ball, D.J. **Hall,"A** novel method of **data** analysis **and Pattern** classification," Springfield, Stanford, **1965,** pp80-85

[3] L. Eltoz, U **Steinbach,and V.** Kumar,"A new **shared** nearest **Neighbor** clusteing algoithm and its applications," AHPCRC, Tech. Rep. 134

[4] Alexander Hinneburg. Daniel **A.** Keim, **"A** General Approach to Clustering in **Large** Databases with ***Noise,"*** Knowledge **and** Information Systems, vol5,2003, pp387-415

[5] M Ester, H.-P. Knegel, I. **Sander, and** X. **Xu, "A** Density-Based **Algorithm** for Discovering Clusters in Large **Spatial** Databass with **Noise," In** p e e d i n g of Knowledge Discovery **and** Data Mining, **1996,pp** 226-231