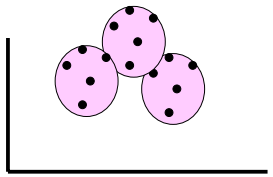
A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases in Noise

Martin Ester, Hans-Peter Kriegel, Jorg Sander, Xiawei Xu

Presenter : Chihoon, Lee
20, Nov, 2000

DBSCAN

1. **Introduction**
2. **Density Based Notion of Clusters**
3. **Overview of DBSCAN**
4. **Performance Evaluation**
5. **Discussion**



Introduction

Spatial Databases

- Require to detect knowledge from great amount of data
- Need to handle with arbitrary shape

Requirements of Clustering in Data Mining

- Scalability
- Dealing with different types of attributes
- Discovery of Clusters with arbitrary shape
- Minimal requirements for domain knowledge to determine input parameters
- Able to deal with noise and outliers
- Insensitive to the order of input data
- High dimensionality of data
- Interpretability and usability

Introduction(cont..)

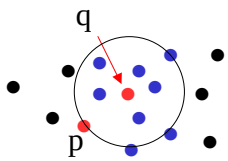
Partitioning	Hierarchical
Domain Knowledge required (K)	Termination Conditions required
K-means (Center) K-medoids(One of Objects) Clarans → Focusing techniques	Agglomerative approach (D_{min}) Divisive approach → Ejcluster $O(n^2)$

Density Based Notion of Clusters

Terms

$N_{eps}(q)$, MinPts

$N_{eps}(q) = \{p \in D \mid dist(p,q) \leq Eps\}$



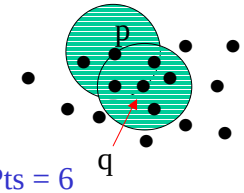
p : border point
q : core point

Definitions

1. Directly Density-reachable
2. Density – reachable
3. Density – connected

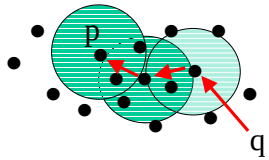
Density Based Notion of Clusters(cont)

Directly Density-reachable

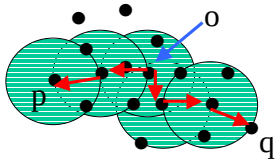


MinPts = 6

Density – reachable



- Cluster : set of density-connected points which is maximal with respect to density-reachability
- Noise – points that don't belong to any Cluster



Density – connected

Overview of DBSCAN

1. Based on Notion of Density in N-dimensional points.
   → Best working with Point data
2. N_{eps} and *MinPts* parameters required.
   → Empirically determined
3. Performed to discover arbitrary shape.
4. Supported by R^* tree structure
   → spatial index structures
   → O(log n)

Overview of DBSCAN (cont..)

Basic 2 steps

1. Arbitrary selection of an point
2. Retrieve all points that are density-reachable

```
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 of DBSCAN
```

Performance Evaluation

- Scalability ✓

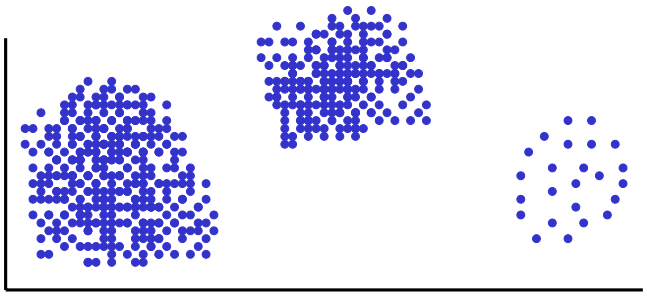
Numbers of points	1252	2503	3910	5213	6256	7820	8937	10426	12512
DBSCAN	3.1	6.7	11.3	16.0	17.8	24.5	28.2	32.7	41.7
CLARANS	758	3026	6845	11745	18029	29826	39265	60540	80638

Performance Evaluation

- Scalability ✓
- Dealing with different types of attributes
- Discovery of Clusters with arbitrary shape ✓
- Able to deal with noise and outliers ✓
- Insensitive to the order of input data ?
- High dimensionality of data ✓
- Interpretability and usability ✓
- Minimum requirements for Domain knowledge to input parameters ✓

Discussion

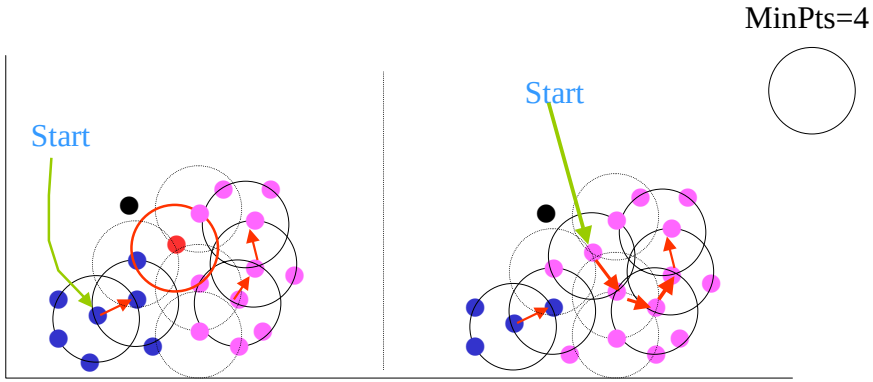
- Requires one Global parameters.



Discussion

- Need to extend object types.
- High Dimensional features need to be investigated.
- Need to explore K-dist graph
- Update clusters for new data

Performance Evaluation(cont)



Performance Evaluation(cont)

Discovered by CLARANS

Discovered by DBSCAN

Overview of DBSCAN (cont..)

2.1. Adopted Heuristic to decide N_{eps} and *MinPts* parameters

Generates K-dist graph
Users or the system estimate percentage of noise
Users can evaluate the selected threshold.