DBSCAN –Density based Clustering

DBSCAN

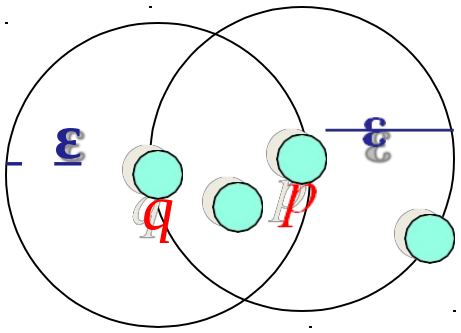
- **Basic idea**
  - Density based locates region of high density that are separated from one another with regions of low density.
  - A cluster is defined as a maximal set of density- connected points
  - Discovers clusters of arbitrary shape based on the notion of density

Density Definition

- ε-Neighborhood – Objects within a radius of ε from a object

$$N_\varepsilon(p) : \{q \mid d(p, q) \leq \varepsilon\}$$

- "High density" - ε-Neighborhood of an object contains at least *MinPts* of objects.

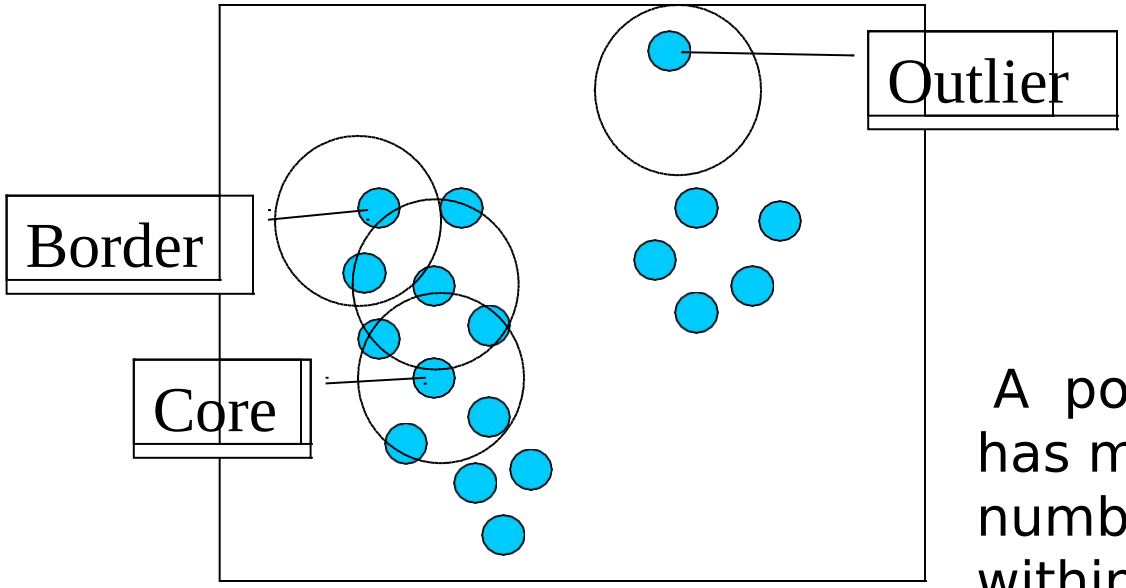


ε-Neighborhood of *p*
ε-Neighborhood of *q*

Density of p is "high" (MinPts = 4)

Density of q is "low" (MinPts = 3)

Core, Border & Outlier



Given ε and *MinPts*, categorize the objects into **three exclusive** groups.

A point is a **core point** if it has more than a specified number of points (MinPts) within Eps—These are points that are at the interior of a cluster.

ε = 1unit,

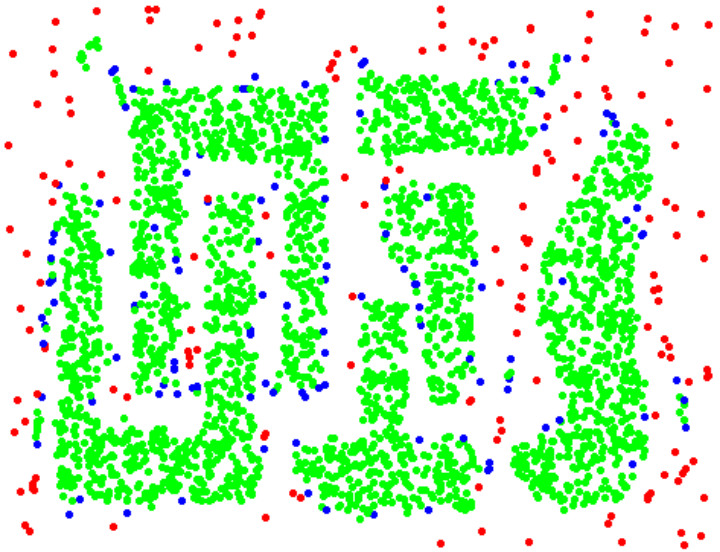
MinPts = 5

A border point has fewer than MinPts within Eps, but is in the neighborhood of a core point.

A noise point is any point that is not a core point nor a border point.

DBSCAN: Core, Border and Noise Points

Original Points

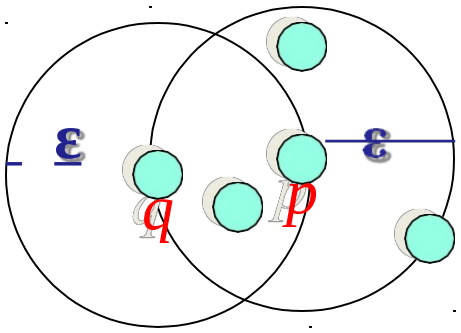


Point types: core, border and noise

Eps = 10, MinPts = 4

Directly Density-reachability

- Directly density-reachable
  - An object *q* is directly density-reachable from object *p*
  - if *p* is a core object and *q* is in p's ε - neighborhood.

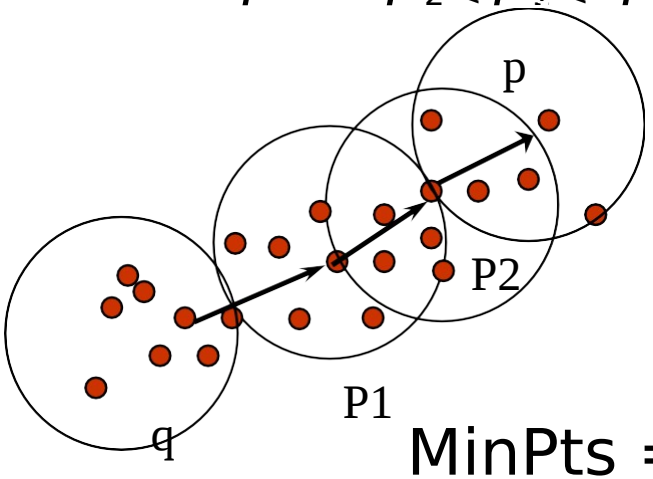


MinPts = 4

- *q* is directly density-reachable from *p*
- *p* is not directly density-reachable from q
- Density-reachability is asymmetric

Indirectly Density-reachability

- Density-Reachable (directly and indirectly):
  - A point p is directly density-reachable from p_2
  - p_2 is directly density-reachable from p_1
  - p_1 is directly density-reachable from q
  - p <- p_2 <- p_1 <- q form a chain

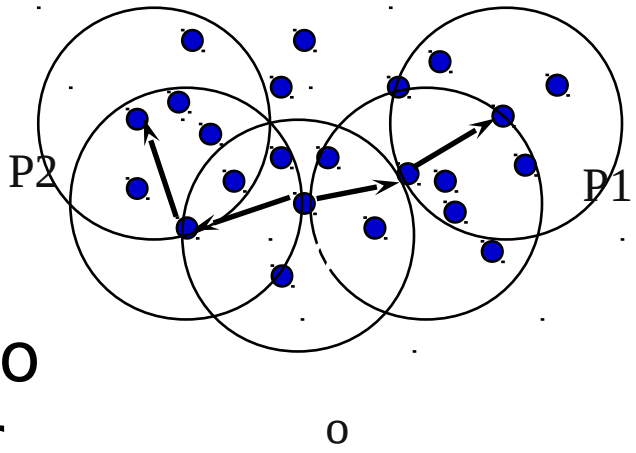


MinPts = 7

- p is (indirectly) density-reachable from q
- q is not density-reachable from p

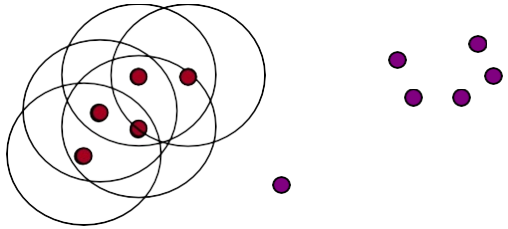
Density Connectedness

- To connect core objects as well as their neighbours in a dense region.
- DBSCAN uses the notion of density-connectedness.
- Two objects p1,p2 belongs to D are density connected w.r to ε and MinPts if there is an object q belongs to D such that both p1 and p2 are density reachable from q w.r to ε and Minpts



DBSCAN Algorithm: Example

- **Parameter**
  - ε = 2 cm
  - *MinPts* = 3

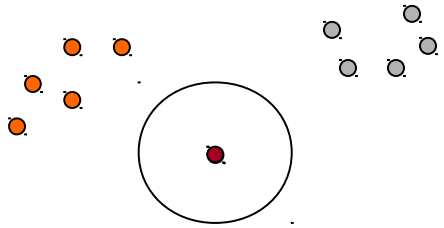


for each *o* ε *D* **do**
if *o* is not yet classified **then** **if** *o* is a core-object **then**
collect all objects density-reachable from *o*
and assign them to a new cluster
else
assign *o* to NOISE

DBSCAN Algorithm: Example

Parameter

- ε = 2 cm
- *MinPts* = 3



for each $o \in D$ **do**
    if o is not yet classified **then**
        if o is a core-object **then**
            collect all objects density-reachable from o
            and assign them to a new cluster.
        else
            assign o to NOISE

DBSCAN Algorithm

Algorithm: DBSCAN: a density-based clustering algorithm.
Input: D: a data set containing n objects, : the radius parameter, and MinPts: the neighborhood density threshold.
Output: A set of density-based clusters.
Method: (1) mark all objects as unvisited;

```
(2) do
(3)                randomly select an unvisited object p;
(4)                mark p as visited;
(5)                if the -neighborhood of p has at least MinPts objects
(6)                  create a new cluster C, and add p to C;
(7)                  let N be the set of objects in the -neighborhood of p;
(8)                  for each point p'  in N
(9)                       if p' is unvisited
(10)                            mark p' as visited;
(11)                            if the -neighborhood of p' has at least MinPts
                                points,add those points to N ;
(12)                  if p' is not yet a member of any cluster, add p' to C;
(13)           end for
(14)           output C;
(15) else mark p as noise;
(16) until no object is unvisited;
```

DBSCAN: Sensitive to Parameters

Figure 8. DBScan results for DS1 with MinPts at 4 and Eps at (a) 0.5 and (b) 0.4.

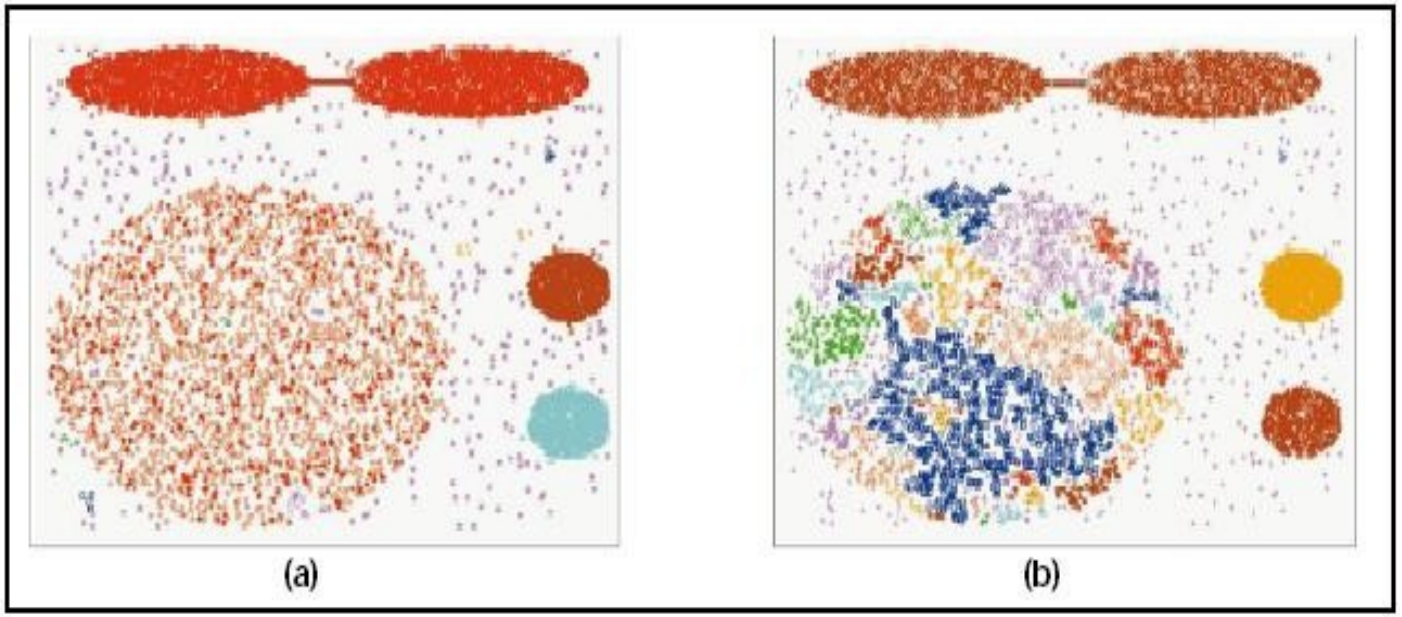
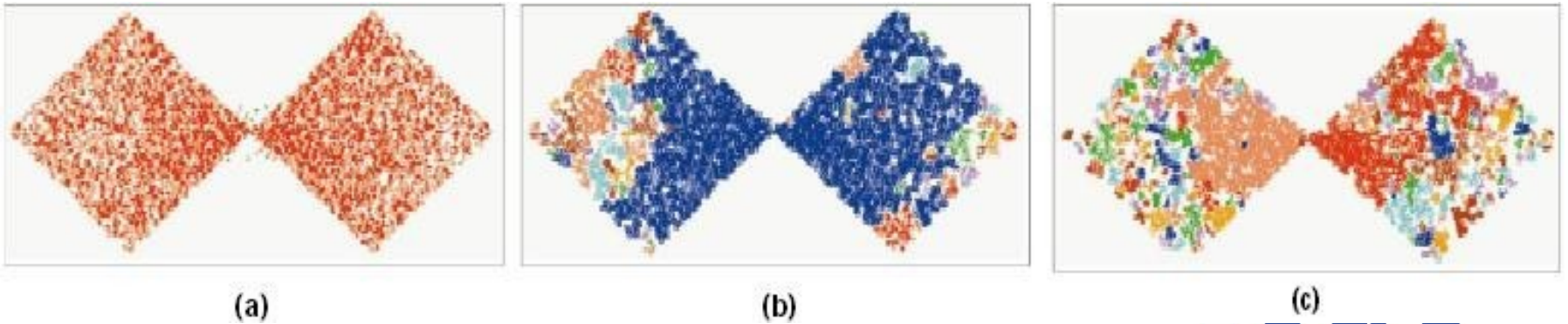
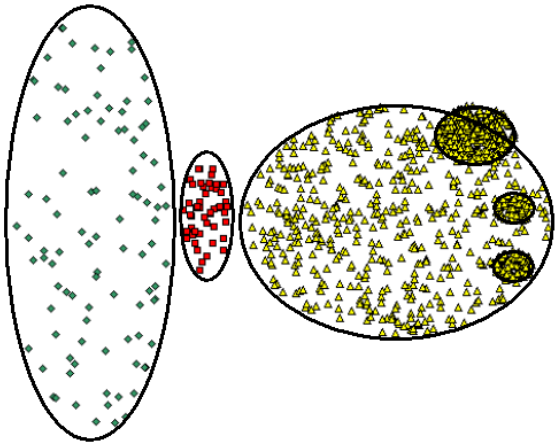


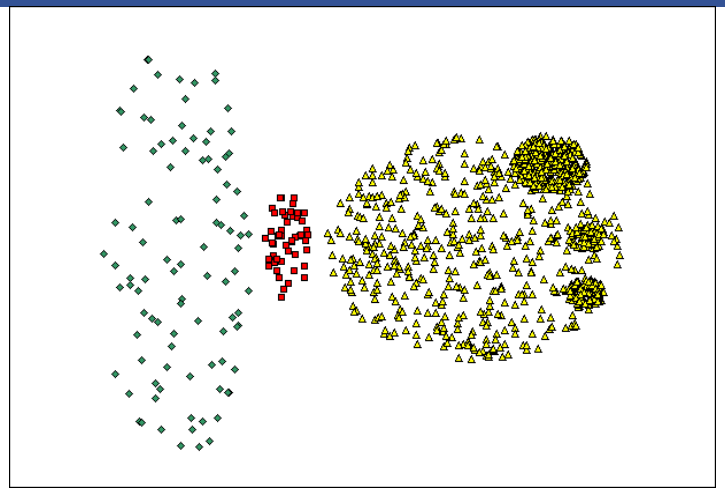
Figure 9. DBScan results for DS2 with MinPts at 4 and Eps at (a) 5.0, (b) 3.5, and (c) 3.0.



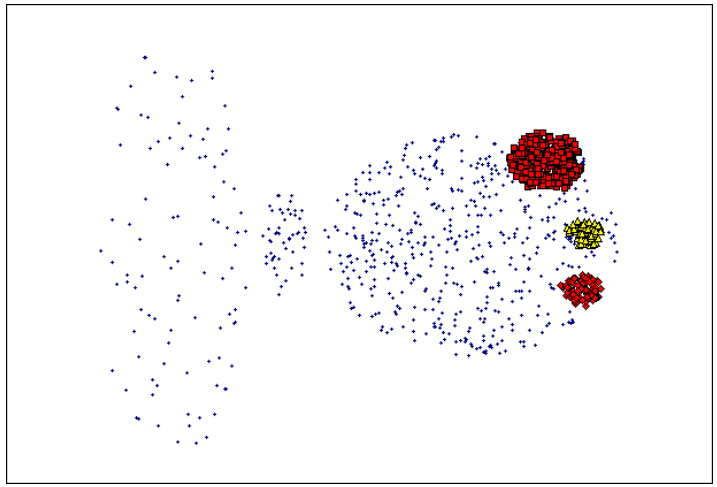
When DBSCAN Does NOT Work Well



Original Points



(MinPts=4, Eps=9.92).



(MinPts=4, Eps=9.75)

- **Cannot handle Varying densities**
- **sensitive to parameters**

OPTICS: Ordering Points To Identify the Clustering Structure

DBSCAN

Input parameter – hard to determine.

Algorithm very sensitive to input parameters.

OPTICS – Ankerst, Breunig, Kriegel, and Sander (SIGMOD'99)

Based on DBSCAN.

Does not produce clusters explicitly.

Rather generate an ordering of data objects representing density-based clustering structure.

OPTICS con't

- Produces a special order of the database wrt its density-based clustering structure
- This cluster-ordering contains info equiv to the density-based clusterings corresponding to a broad range of parameter settings
- Good for both automatic and interactive cluster analysis, including finding intrinsic clustering structure
- Can be represented graphically or using visualization techniques

SSN

OPTICS: Extension of DBSCAN

- To construct the different clusterings simultaneously the objects are processed in a specific order.
- The order selects an object that is density reachable w.r to lowest Eps so that clusters w.r.t. higher density are finished first.
- OPTICS needs two pieces of information per object
    - Core-distance
    - Reachability distance

Core- and Reachability Distance

Parameters: "generating" distance ε, fixed value MinPts

core-distance$_{\varepsilon,MinPts}$(o)

- *"smallest distance ε' such that o is a core object and ε' neighborhood o has atleast MinPts objects*
  (if that distance ε' is $\leq \varepsilon$ that makes o as core object otherwise o is undefined*)*

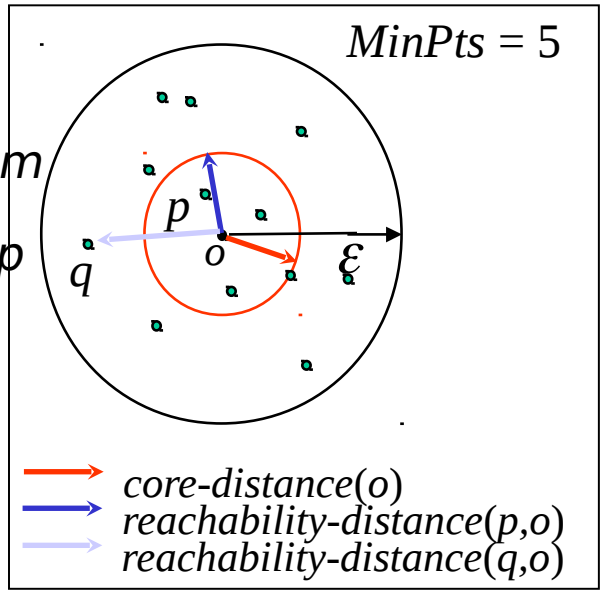
reachability-distance$_{\varepsilon,MinPts}$(p, o)

The reachability distance from p to o is the minimum

Radius value that makes o density reachable from p

The reachability distance from p to o is

max{core_distance (o),dist(o,p))



Core- and Reachability Distance

- It computes an ordering of all objects in a given database. And
- It stores the core-distance and a suitable reachability-distance for each object in the database.
- OPTICS maintains a list called OrderSeeds to generate the output ordering.
- Objects in OrderSeeds
  - are sorted by the reachability-distance from their respective closest core objects,
  - that is, by the smallest reachability-distance of each object.

Core- and Reachability Distance

- Begin with an arbitrary object from the input database as the current object, p.
- It retrieves the ε-neighborhood of p, determines the core-distance, and sets the reachability-distance to undefined.
- The current object, p, is then written to output.
- If p is not a core object,
  - OPTICS simply moves on to the next object in the OrderSeeds list (or the input database if OrderSeeds is empty).

Core- and Reachability Distance

- If p is a core object,
  - then for each object, q, in the ε-neighborhood of p,
  - OPTICS updates its reachability-distance from p
  - and inserts q into OrderSeeds if q has not yet been processed.
- The iteration continues until the input is fully consumed and OrderSeeds is
- empty.

OPTICS: The Reachability Plot

- *represents the density-based clustering structure*
- *easy to analyze*
- *independent of the dimension of the data*