Big Data Summer School

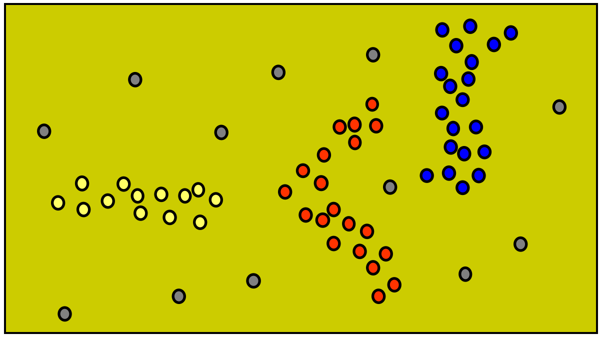
Density-based Approaches

- Density
  - the volume (the number of objects) per unit
- Why Density-Based Clustering methods?
  - Discover clusters of arbitrary shape with noises.
  - Clusters
    - Dense regions of objects separated by regions of low density
- DBSCAN – the first density based clustering

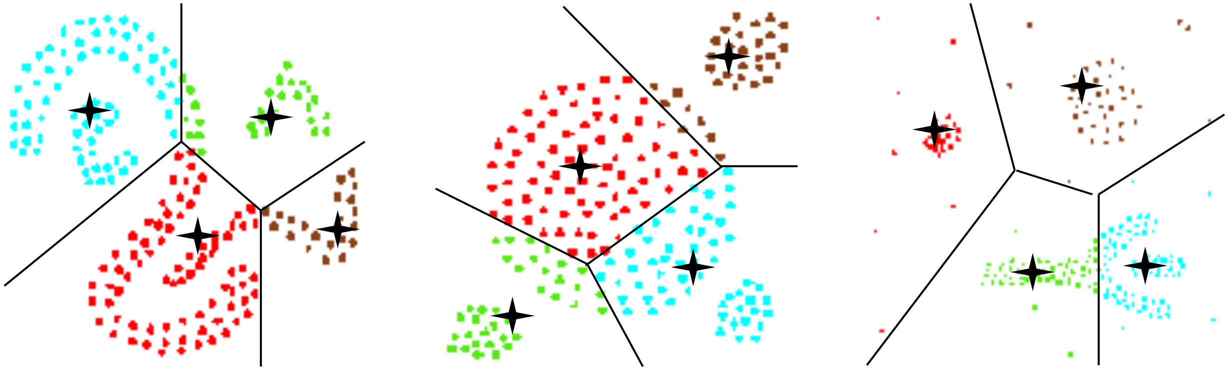
Density-Based Clustering

- ***Basic Idea*:**

  Clusters are dense regions in the data space, separated by regions of lower object density



- Why Density-Based Clustering?



Results of a *k*-medoid algorithm for *k*=4

Density Based Clustering: Basic Concept

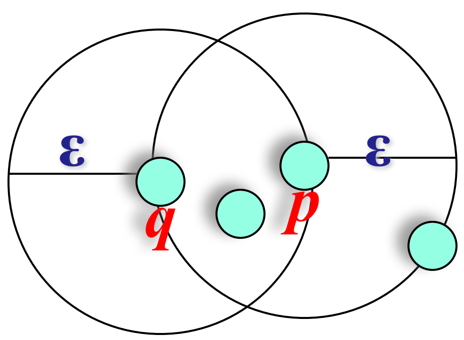
- Intuition for the formalization of the basic idea
  - For any point in a cluster, the local point density around that point has to exceed some threshold
  - The set of points from one cluster is spatially connected
- Local point density at a point *p* defined by two parameters
  - ε – radius for the neighborhood of point p: $N_\varepsilon(p) := \{q \text{ in data set } D \mid dist(p, q) <= \varepsilon\}$
  - *MinPts* – minimum number of points in the given neighbourhood *N*(*p*)

ε-Neighborhood

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

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

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

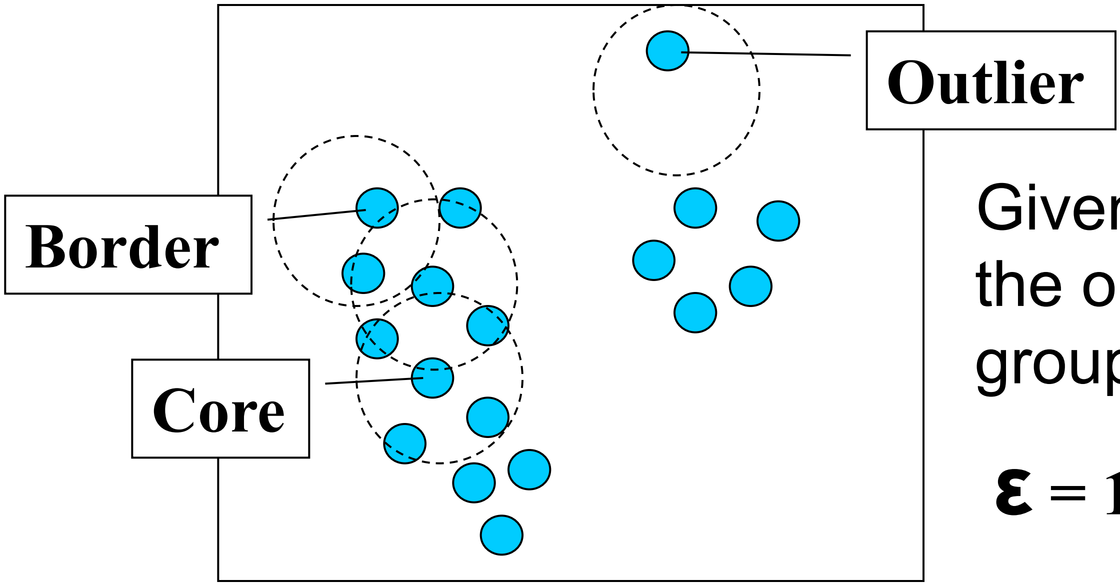


ε-Neighborhood of *p*
ε-Neighborhood of *q*

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

***Density of q* is "low" (MinPts = 4)**

Core, Border & Outlier



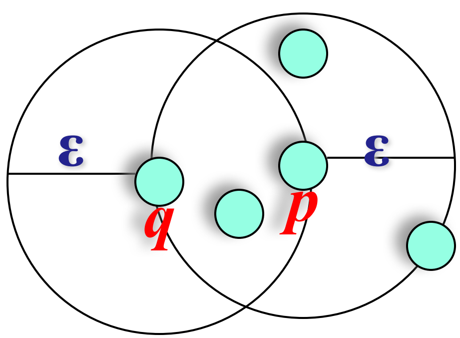
Given *ε* and *MinPts*, categorize the objects into three exclusive groups.

ε = 1unit, MinPts = 5

- A point is a core point if it has more than MinPts within **ε**. Interior of a cluster.
- 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.

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.**

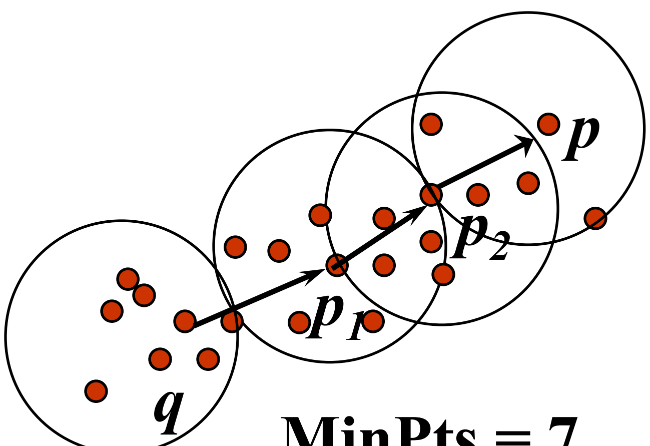


MinPts = 4

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

Density-reachability

- Density-Reachable (directly and indirectly):
  - p is directly density-reachable from p2;
  - p2 is directly density-reachable from p1;
  - p1 is directly density-reachable from q;
  - p←p2←p1←q form a chain.

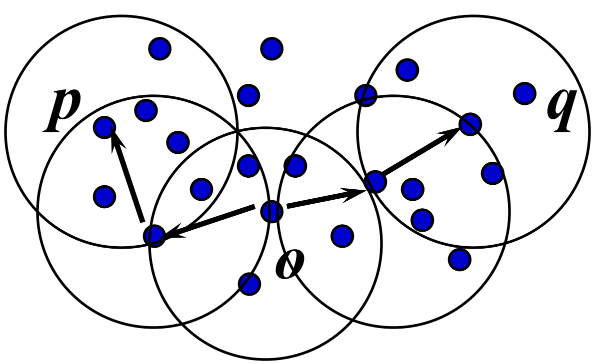


MinPts = 7

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

Density-Connectivity

- **Density-reachable is not symmetric**
  - **not good enough to describe clusters**
- **Density-Connected**
  - **A pair of points p and q are density-connected if they are commonly density-reachable from a point o.**



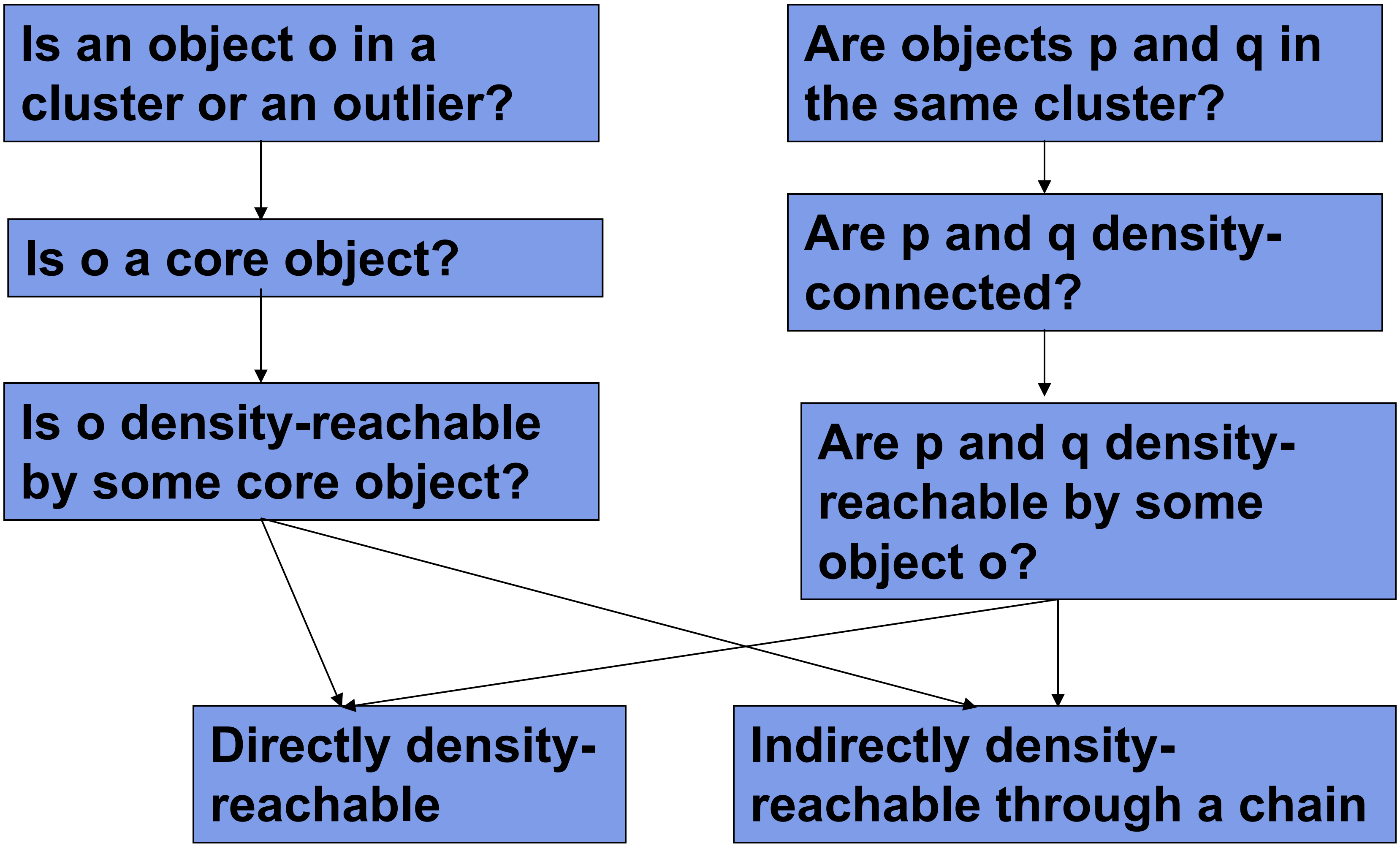
- **Density-connectivity is symmetric**

Formal Description of Cluster

- Given a data set D, parameter ε and threshold MinPts.
- A cluster C is a subset of objects satisfying two criteria:
  - *Connected:* forall p,q in C: p and q are density-connected.
  - *Maximal:* forall p,q: if p in C and q is <u>density-reachable from p</u>, then q in C. (avoid redundancy)

P is a core object.

Review of Concepts



DBSCAN Algorithm

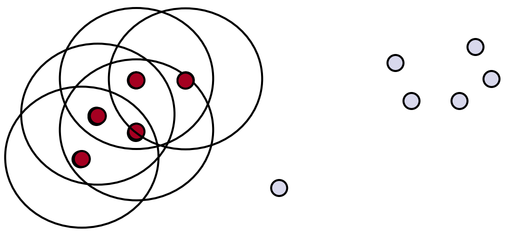
```
Input: The data set D

Parameter: ε, MinPts

For each object p in D
    if p is a core object and not processed then
        C = retrieve all objects density-reachable from p
        mark all objects in C as processed
        report C as a cluster
    else mark p as outlier
    end if

End For
```

DBSCAN: The Algorithm

- 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 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*

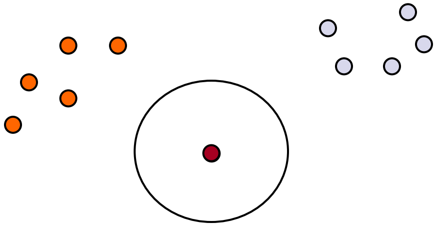
and assign them to a new cluster.

else

assign *o* to NOISE

DBSCAN Algorithm: Example

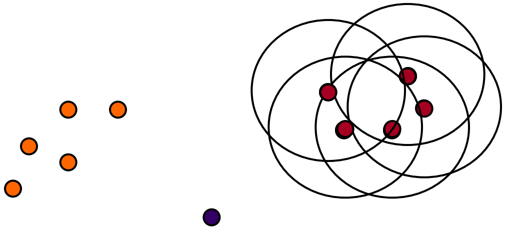
- Parameter
  - ε = 2 cm
  - *MinPts* = 3



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: Example

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

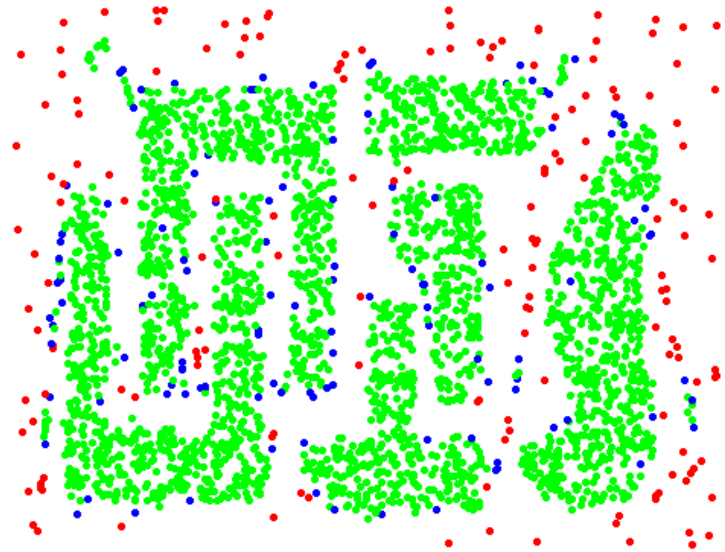


for each *o in D* do
    if *o* is not yet classified then
        if *o* is a core-object then
            collect all objects density-reachable from *o*
            and assign them to a new cluster.
        else
            assign *o* to NOISE

Example

Original Points

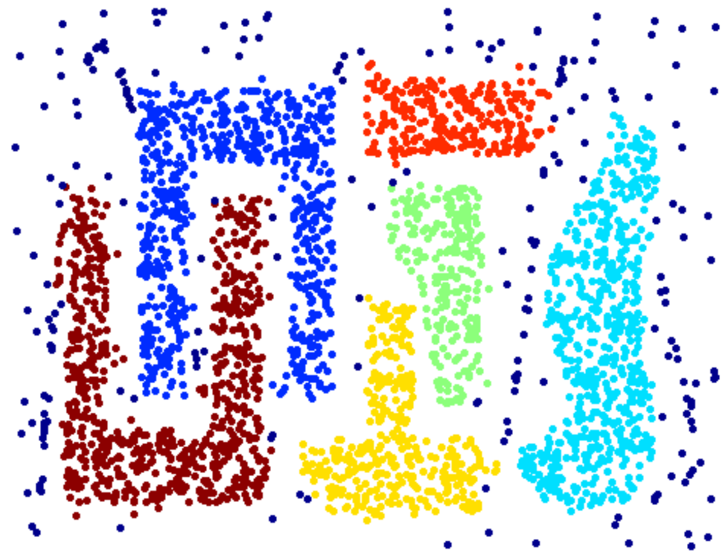
ε = 10, MinPts = 4



Point types: core, border and outliers

When DBSCAN Works Well

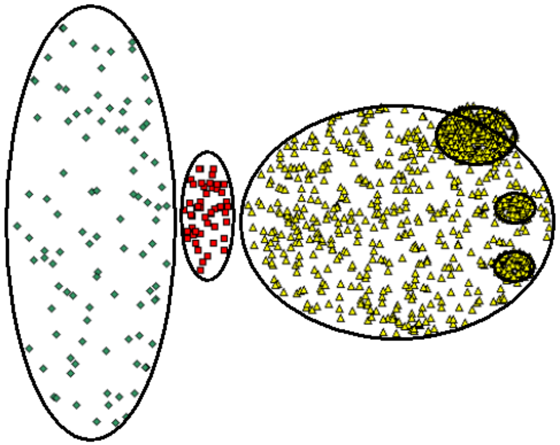
Original Points

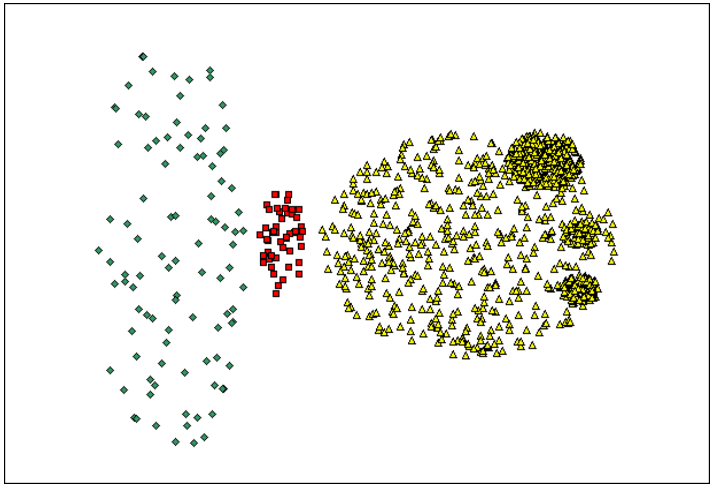


Clusters

- Resistant to Noise
- Can handle clusters of different shapes and sizes

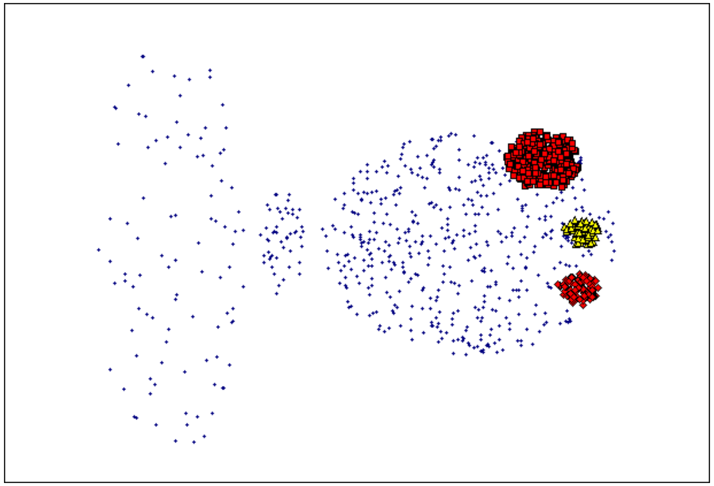
When DBSCAN Does NOT Work Well





(MinPts=4, Eps=9.92).

Original Points

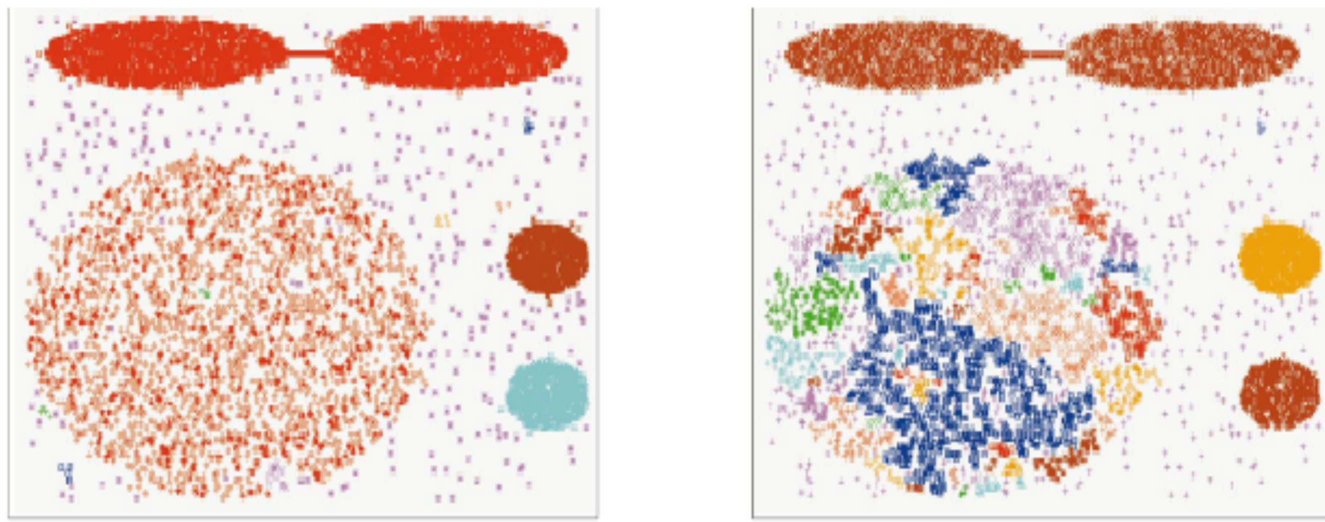


(MinPts=4, Eps=9.75)

- Cannot handle Varying densities
- sensitive to parameters

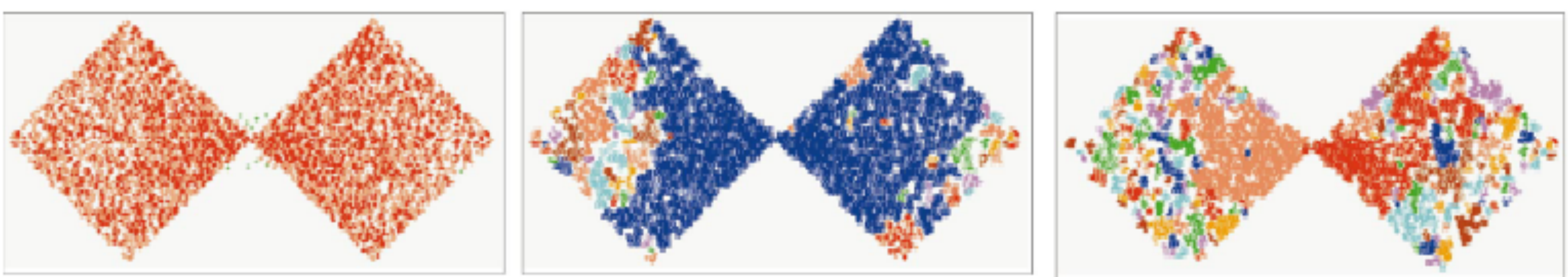
DBSCAN: Sensitive to Parameters

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



(a) (b)

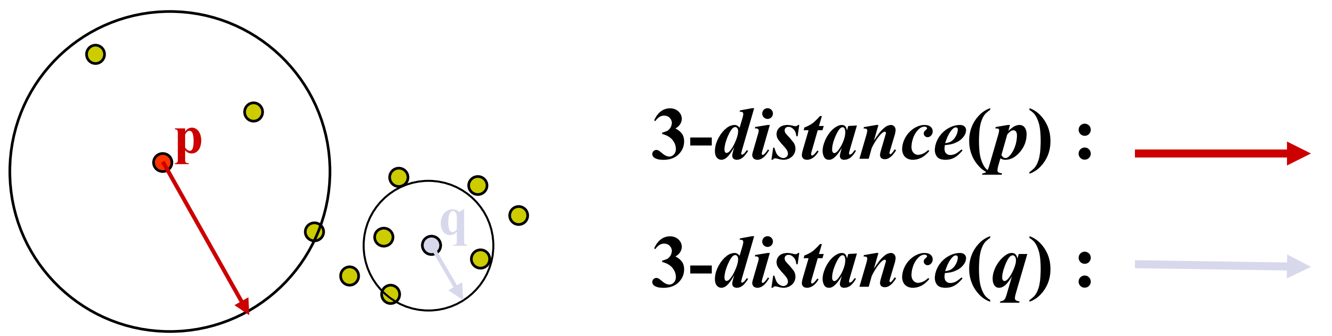
Figure 9. DBScan results for DS2 with MinPts at 4 and Eps at (a) 5.0, (b) 3.5, and (c) 3.0.



(a) (b) (c)

Determining the Parameters *ε* and *MinPts*

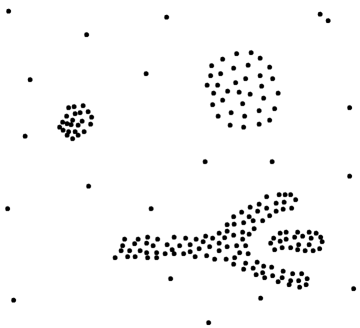
- Cluster: Point density higher than specified by ε and *MinPts*
- MinPts = D+1;
- Heuristic: look at the distances to the *k*-nearest neighbors

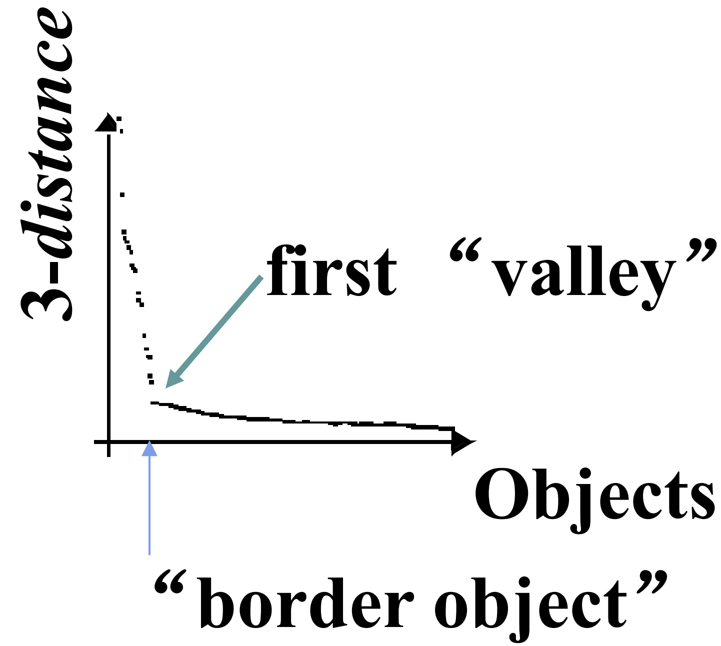


- Function *k-distance*(*p*): distance from *p* to the its *k*-nearest neighbor
- *k-distance plot*: *k*-distances of all objects, sorted in decreasing order

Determining the Parameters ε and *MinPts*

- Example *k*-distance plot

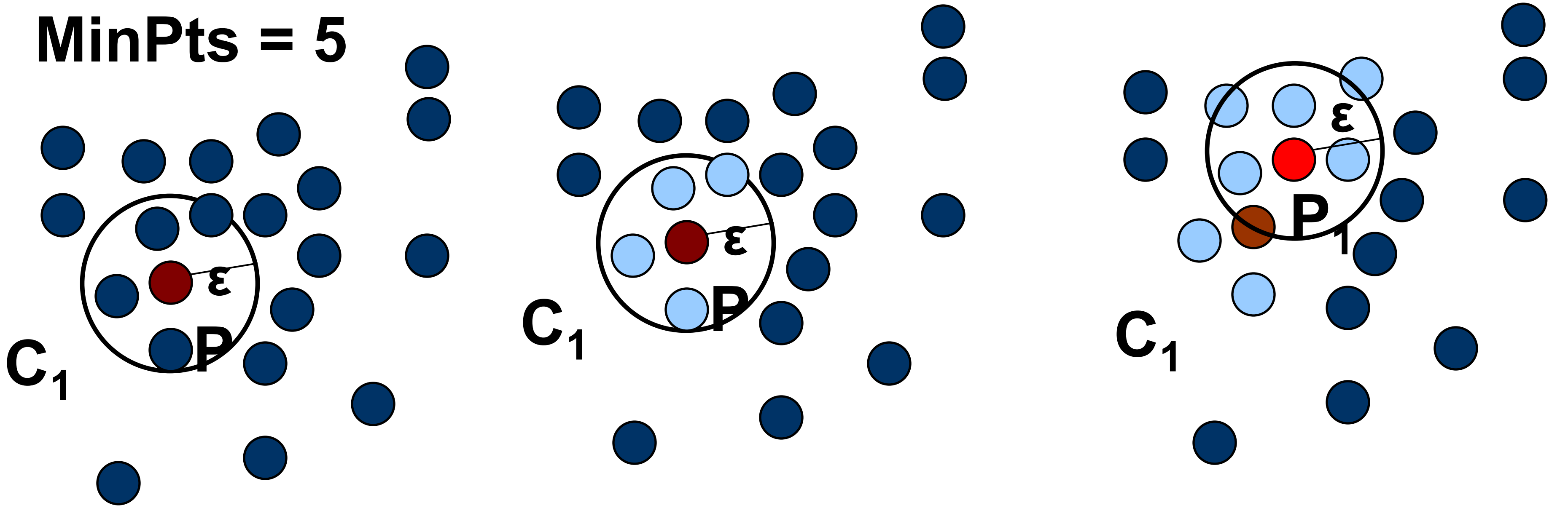




- Heuristic method:
  - Fix a value for *MinPts*
  - User selects “border object” *o* from the *MinPts-distance* plot; ε is set to *MinPts-distance*(o)

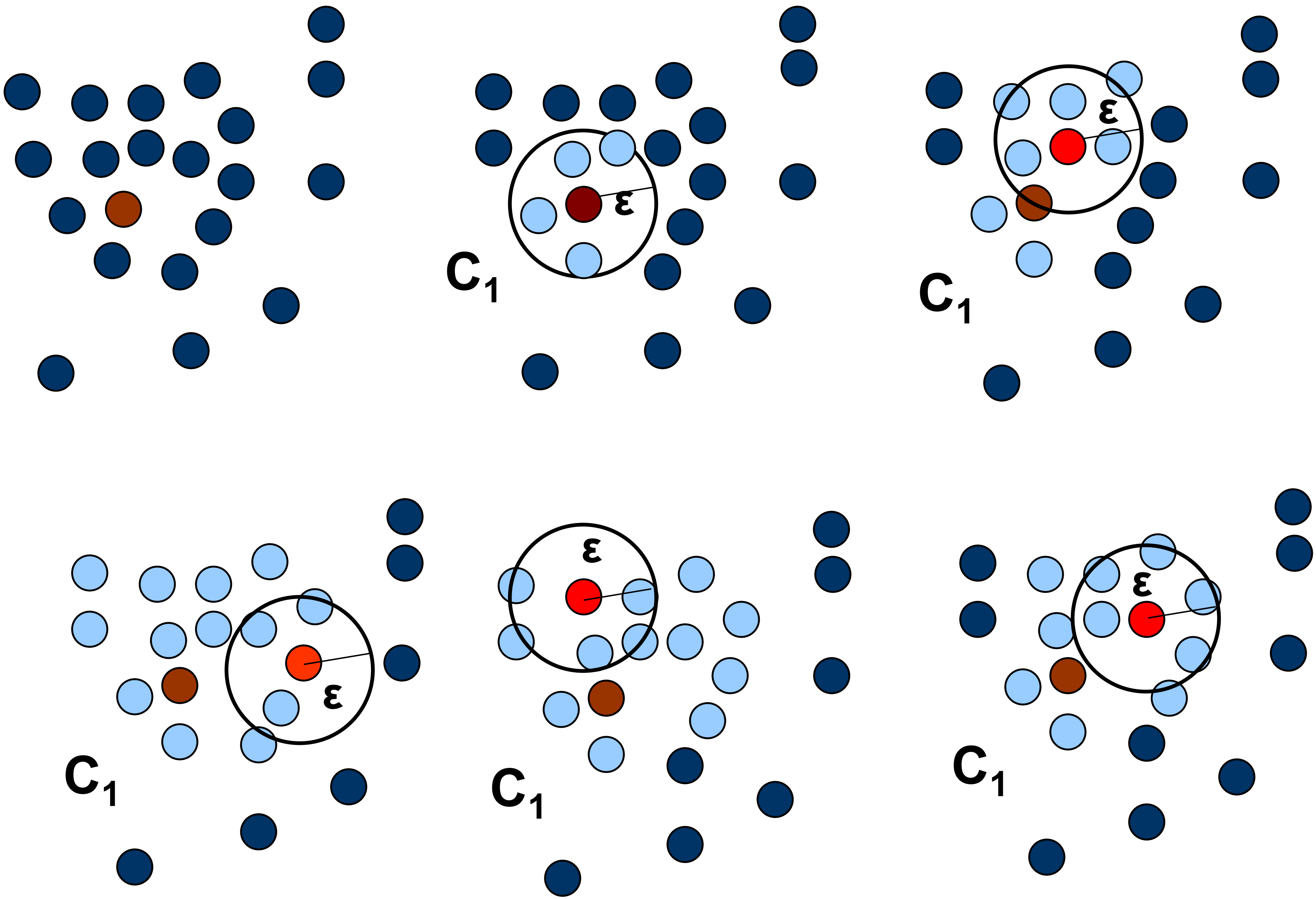
Density Based Clustering: Discussion

- Advantages
  - Clusters can have arbitrary shape and size
  - Number of clusters is determined automatically
  - Can separate clusters from surrounding noise
  - Can be supported by spatial index structures
- Disadvantages
  - Input parameters may be difficult to determine
  - In some situations very sensitive to input parameter setting



1. Check the ε-neighborhood of p;
2. If p has less than MinPts neighbors then mark p as outlier and continue with the next object
3. Otherwise mark p as processed and put all the neighbors in cluster C

1. Check the unprocessed objects in C
2. If no core object, return C
3. Otherwise, randomly pick up one core object p_1, mark p_1 as processed, and put all unprocessed neighbors of p_1 in cluster C



ε
C1
ε
C1
ε
C1
ε
C1
ε
C1