# Outline

- Partition-based clustering
- Hierarchical clustering
- Density-based clustering
- Model-based clustering

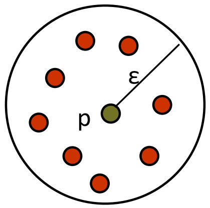
# Density based clustering

- Clusters are regions of high density surrounded by regions of low density (noise)
- Clustering based on density (local cluster criterion), such as density-connected points
- Major features:
  - Discover clusters of arbitrary shape
  - Handle noise
  - One scan
  - Need density parameters as termination condition

- Several interesting studies:
  - DBSCAN: Ester, et al. (KDD'96)
  - OPTICS: Ankerst, et al (SIGMOD'99).
  - DENCLUE: Hinneburg & D. Keim (KDD'98)
  - CLIQUE: Agrawal, et al. (SIGMOD'98) (more grid-based)

# The notion of density

- Density:
  - Density is measured locally in the Eps-neighborhood (or ε-neighborhood) of each point
  - Density = number of points within a specified radius Eps (point itself included)



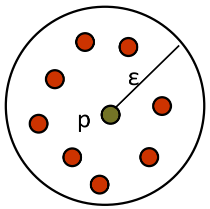

The e-neighborhood of p: 9 points

- Density depends on the specified radius
  - In an extreme small radius, all points will have a density of 1 (only themselves)
  - In an extreme large radius, all points will have a density of N (the size of the dataset)

# DBSCAN basic concepts

- Consider a dataset D of objects to be clustered
- Two parameters:
  - Eps (or ε): Maximum radius of the neighbourhood
  - MinPts: Minimum number of points in an Eps-neighbourhood of that point

- Eps-neighborhood of a point p in D
  - $N_{Eps}(p)$: {q belongs to D | dist(p,q) <= Eps}




The Eps-neighborhood of p

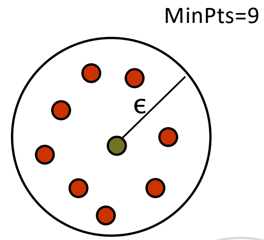
# Core points vs border points vs noise points

- Let D be a dataset. Given a radius parameter Eps and a density parameter MinPts we can distinguish between:
  - Core points

    A point is a core point if it has more than a specified number of points (MinPts) within a specified radius Eps, i.e.,:

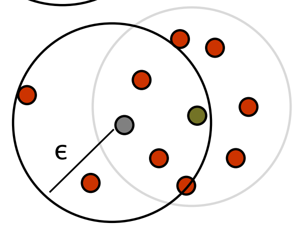
    $$|N_{Eps}(p)=\{q \mid dist(p,q) <= Eps \}| \geq MinPts$$

    - These are points that are at the interior of a cluster



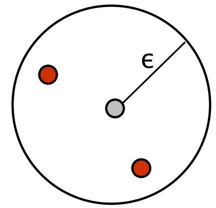

  - Border points

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

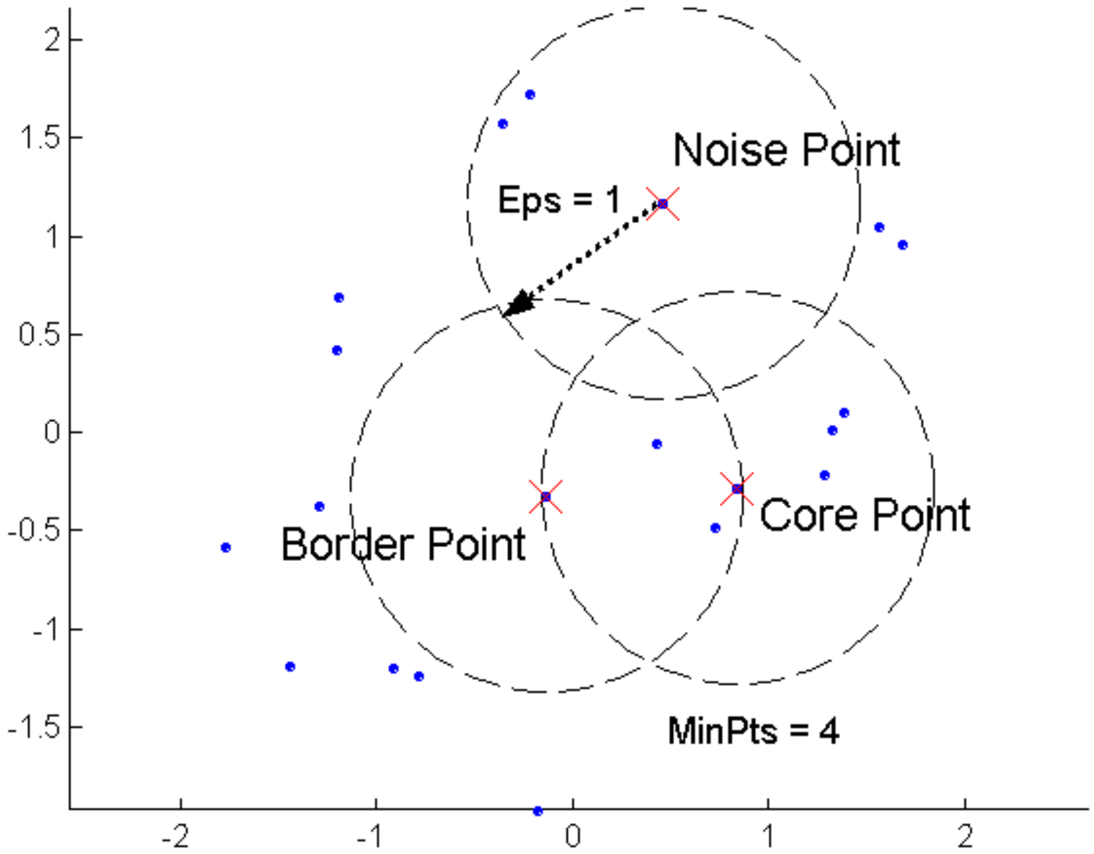



  - Noise points

    neither a core point nor a border point.

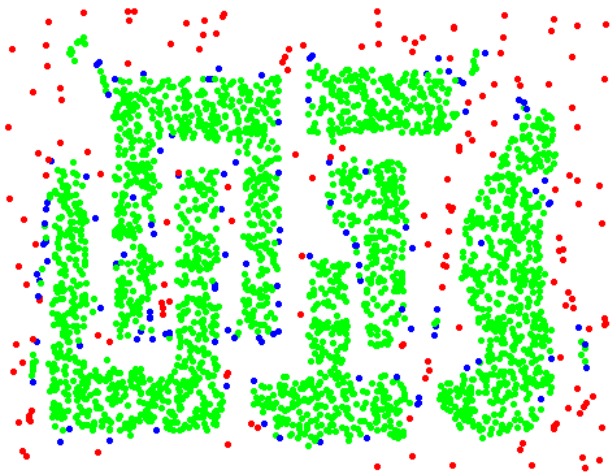


# Example


Noise Point
Eps = 1
Border Point
Core Point
MinPts = 4
2
1.5
1
0.5
0
-0.5
-1
-1.5
-2
-1
0
1
2

# Core, Border and Noise points

Original points

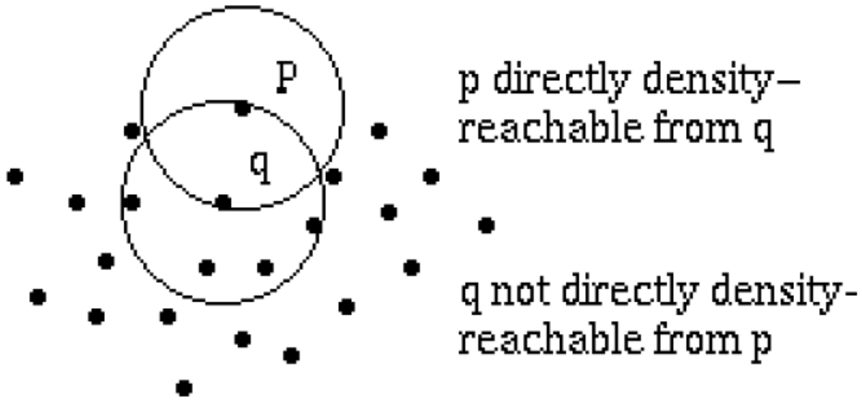


Point types: core, border and noise

Eps = 10, MinPts = 4

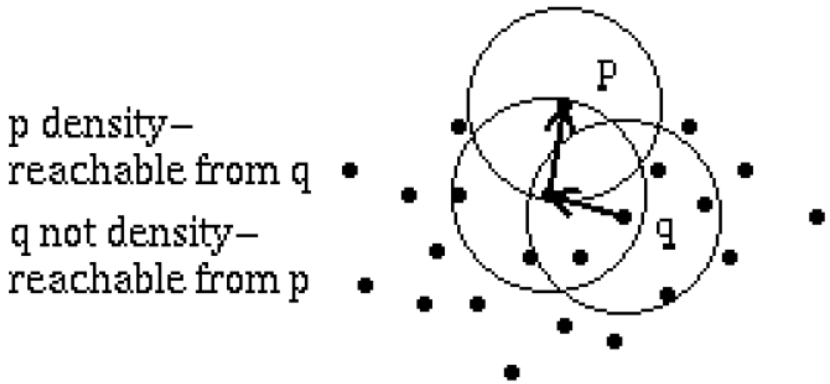
# Direct reachability

- Directly density-reachable: A point *p* is directly density-reachable from a point *q* w.r.t. *Eps*, *MinPts* if
  - *p* belongs to $N_{Eps}(q)$
  - q is a core point, i.e.,: $|N_{Eps}(q)| >= MinPts$



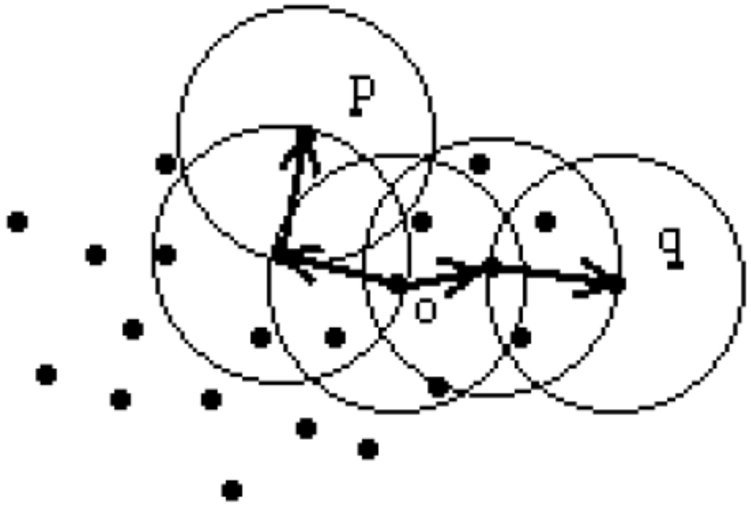
# Reachability

- Density-reachable:
  - A point $p$ is density-reachable from a point $q$ w.r.t. *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 $p_i$



# Connectivity

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



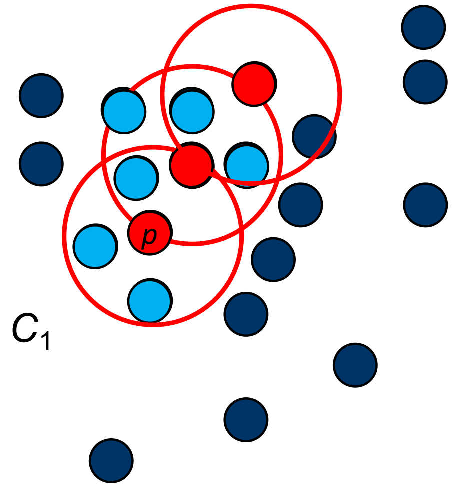

p and q density-connected to each other by o

# Cluster

- A cluster is a maximal set of density-connected points

# DBSCAN algorithm (from Lecture 11)

- Arbitrary select a point *p*
- Retrieve all points density-reachable from *p* w.r.t. *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 pseudocode I

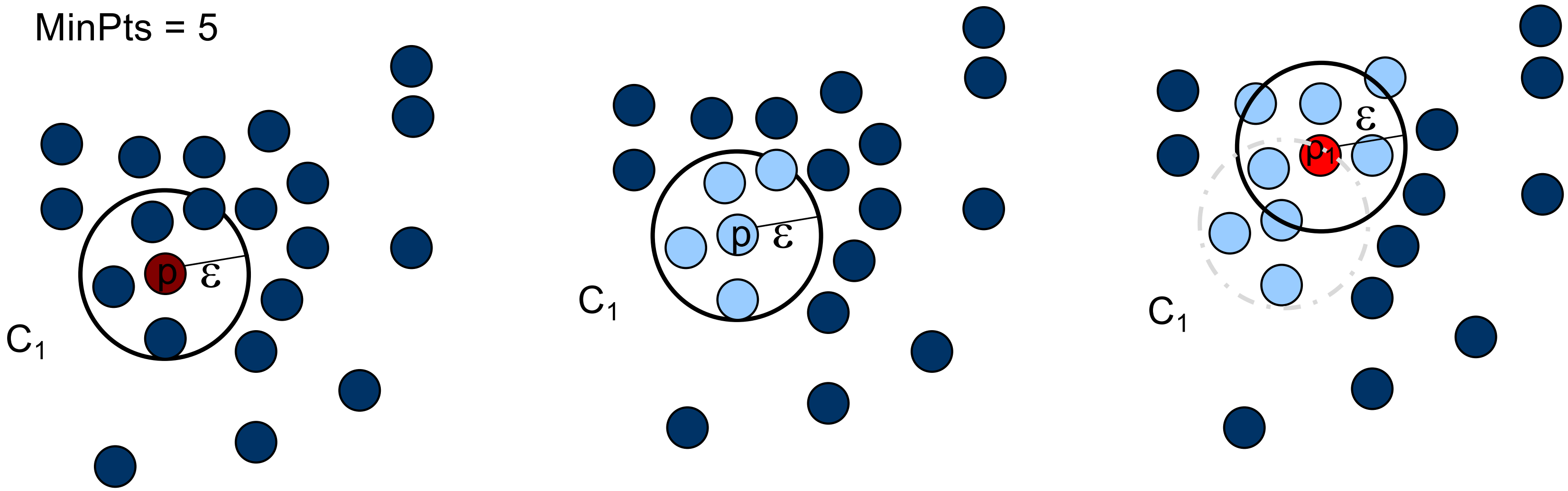
```
DBSCAN(Dataset DB, Real Eps, Integer MinPts)
   // initially all objects are unclassified,
   // o.ClId = unclassified for all o ∈ DB

  ClusterId := nextId(NOISE);
  for i from 1 to |DB| do
     Object := DB.get(i);
     if Object.ClId = unclassified then
        if ExpandCluster(DB, Object, ClusterId, Eps, MinPts)
        then ClusterId:=nextId(ClusterId);
```

# DBSCAN pseudocode II

```
ExpandCluster(DB, StartObject, ClusterId, Eps, MinPts): Boolean
 seeds:= RQ(StartObjekt, Eps);
 if |seeds| < MinPts then // StartObject is not a core object
    StartObject.ClId := NOISE;
        return false;
 else // else: StartObject is a core object
        forall o ∈ seeds do o.ClId := ClusterId;
        remove StartObject from seeds;
    while seeds ≠ Empty do
        select an object o from the set of seeds;
                Neighborhood := RQ(o, Eps);
                if |Neighborhood| ≥ MinPts then // o is a core object
                           for i from 1 to |Neighborhood| do
                               p := Neighborhood.get(i);
                               if p.ClId in {UNCLASSIFIED, NOISE} then
                                  if p.ClId = UNCLASSIFIED then
                                     add p to the seeds;
                                   p.ClId := ClusterId;
                end if;
            end for;
                end if;
                remove o from the seeds;
    end while;
 end if
return true;
```

# DBSCAN: An example*

MinPts = 5




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 $C_1$

1. Check the unprocessed objects in $C_1$
2. If no core object, return $C_1$
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 $C_1$

*Source: http://www.cse.buffalo.edu/faculty/azhang/cse601/density-based.ppt*

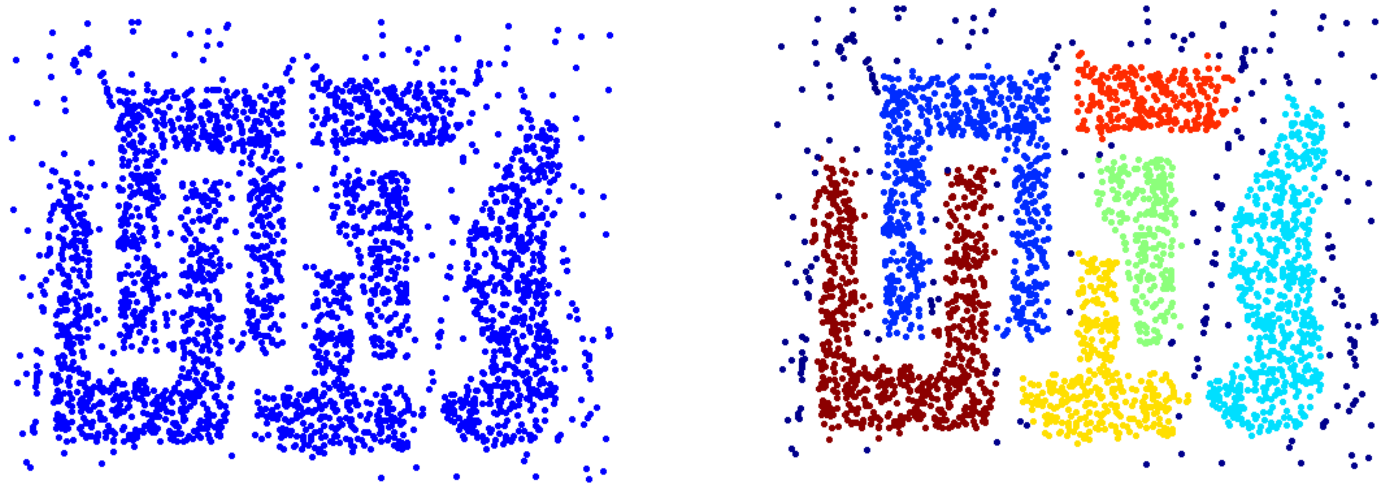
# Complexity

- For a dataset D consisting of n points, the time complexity of DBSCAN is O(n x time to find points in the Eps-neighborhood)

- Worst case $O(n^2)$

- In low-dimensional spaces O(nlogn);
  - efficient data structures (e.g., *kd-trees*) allow for efficient retrieval of all points within a given distance of a specified point

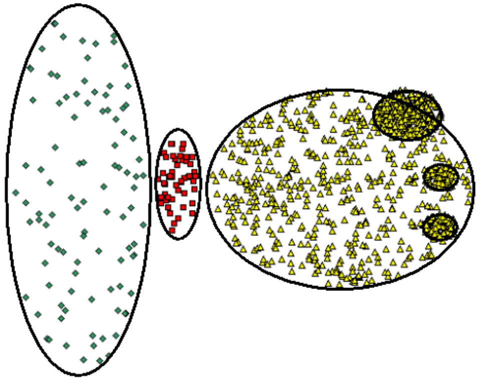
# When DBSCAN works well?



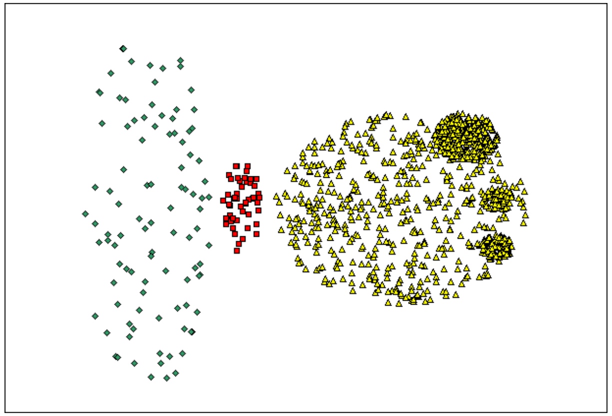
Clusters

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

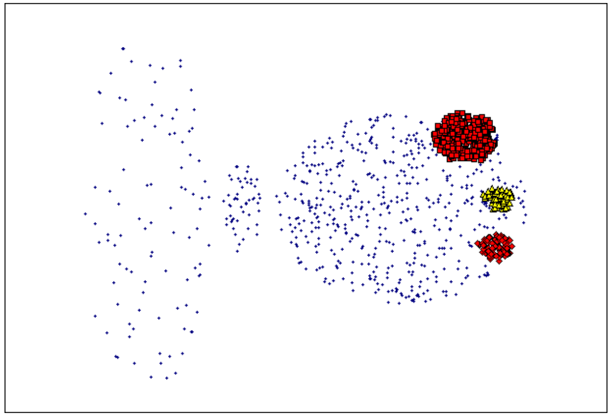
# When DBSCAN does not work well?



Original points



(MinPts=4, Eps=9.92).

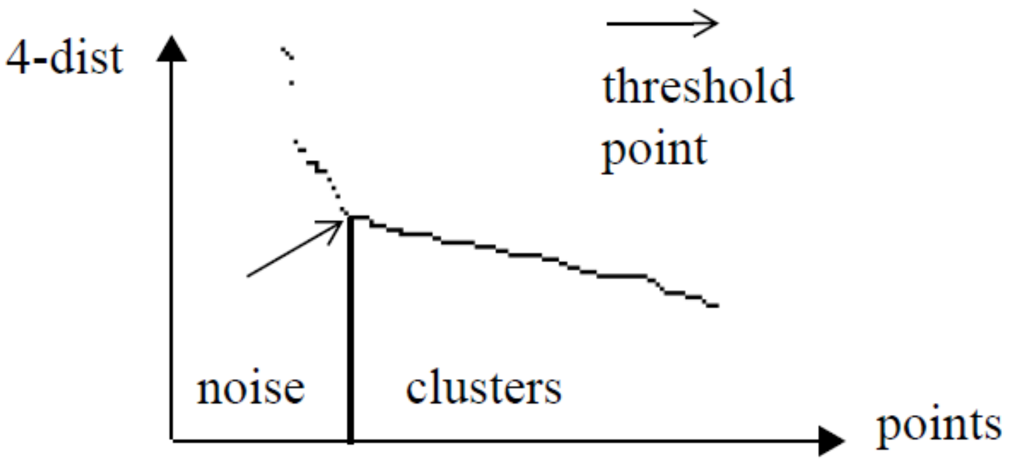


(MinPts=4, Eps=9.75)

- Varying densities
- High-dimensional data

## DBSCAN: determining Eps and MinPts

- Idea is that for points in a cluster, their $k^{th}$ nearest neighbors are at roughly the same distance
- Noise points have the $k^{th}$ nearest neighbor at farther distance
- So, plot sorted distance of every point to its $k^{th}$ nearest neighbor




Ordering points to identify the clustering structure (OPTICS algorithm)