

DBSCAN

Density-based Spatial Clustering of Applications with Noise

Faculty of Mathematics and Computer Science, University of Bucharest
and
Sparktech Software

Academic Year 2018/2019, 1st Semester

DBSCAN

- **DBSCAN** is a *density-based* clustering algorithm which groups points that are *closely packed* together in feature space.
 - “A density-based algorithm for discovering clusters in large spatial databases with noise”

Martin Ester , Hans-Peter Kriegel , Jörg Sander , Xiaowei Xu, 1996

- Unlike *K-means*, it does not require the *number of clusters* to be known in advance, but it has other *hyperparameters* which define the density of the clusters it looks for.

Terminology

- Let X be a dataset of points.
- **ϵ -neighborhood** of a point p is the set of points within an area of radius ϵ around the point.

$$N_{\epsilon}(p) = \{q \in X \mid \text{dist}(p, q) \leq \epsilon\}$$

Terminology

- Let X be a dataset of points.
- ϵ -**neighborhood** of a point p is the set of points within an area of radius ϵ around the point.

$$N_{\epsilon}(p) = \{q \in X \mid \text{dist}(p, q) \leq \epsilon\}$$

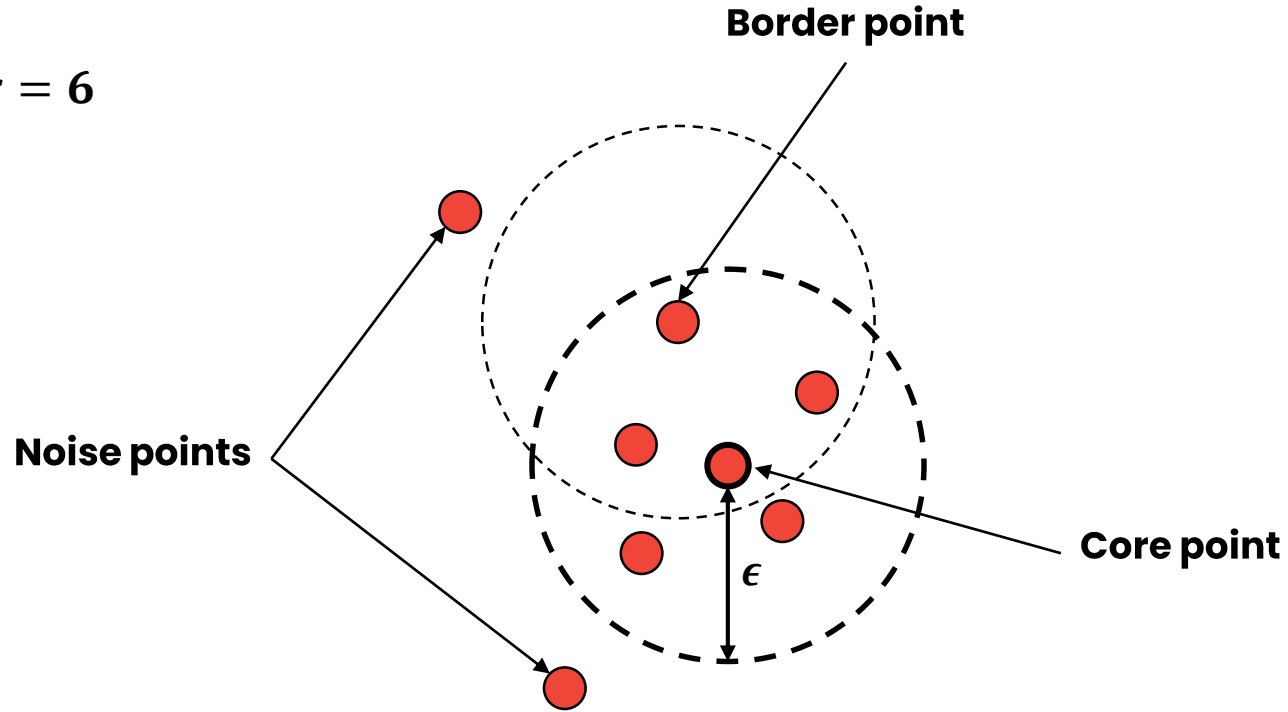
- A point p is a **core point** if its ϵ -neighborhood has at least $MinPts$ points.

$$|N_{\epsilon}(p)| \geq MinPts$$

- p is a **border point** if it is in the ϵ -neighborhood of a *core point*, but it is not itself a core point.
- All points which are neither *core points* nor *border points* are **noise points**.

Terminology

MinPts = 6



Terminology

- A point q is **directly density-reachable** from a point p if p is a *core point* and q is in the ϵ -neighborhood of p .

$$|N_\epsilon(p)| \geq \text{MinPts} \text{ and } q \in N_\epsilon(p)$$

Terminology

- A point q is **directly density-reachable** from a point p if p is a *core point* and q is in the ϵ -neighborhood of p .

$$|N_\epsilon(p)| \geq \text{MinPts} \text{ and } q \in N_\epsilon(p)$$

- q is **density-reachable** from p if there is a chain of points $p_1, p_2, \dots, p_n \in X$ with $p_1 = p$, $p_n = q$ such that p_{i+1} is *directly density-reachable* from p_i
- p and q are **density-connected** if there is a point $o \in X$ such that both p and q are *density-reachable* from o .

Terminology

- A point q is **directly density-reachable** from a point p if p is a *core point* and q is in the ϵ -neighborhood of p .

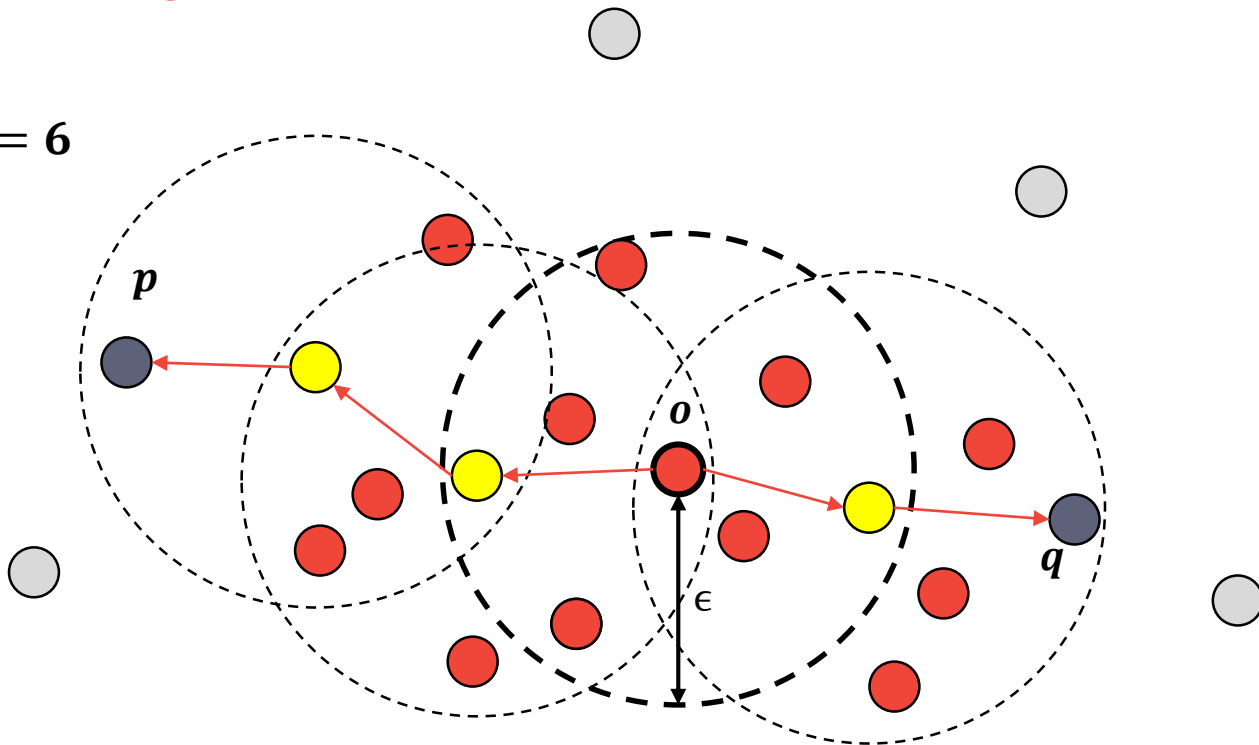
$$|N_\epsilon(p)| \geq \text{MinPts} \text{ and } q \in N_\epsilon(p)$$

- q is **density-reachable** from p if there is a chain of points $p_1, p_2, \dots, p_n \in X$ with $p_1 = p$, $p_n = q$ such that p_{i+1} is *directly density-reachable* from p_i
- p and q are **density-connected** if there is a point $o \in X$ such that both p and q are *density-reachable* from o .
- A **cluster** $C \subset X$ is a set of points which satisfies two conditions:
 - **Maximality:** $\forall p, q \in X$, if $p \in C$ and q is *density-reachable* from $p \Rightarrow q \in C$
 - **Connectivity:** $\forall p, q \in C$, p and q are *density-connected*

A cluster contains both core and border points.

Terminology

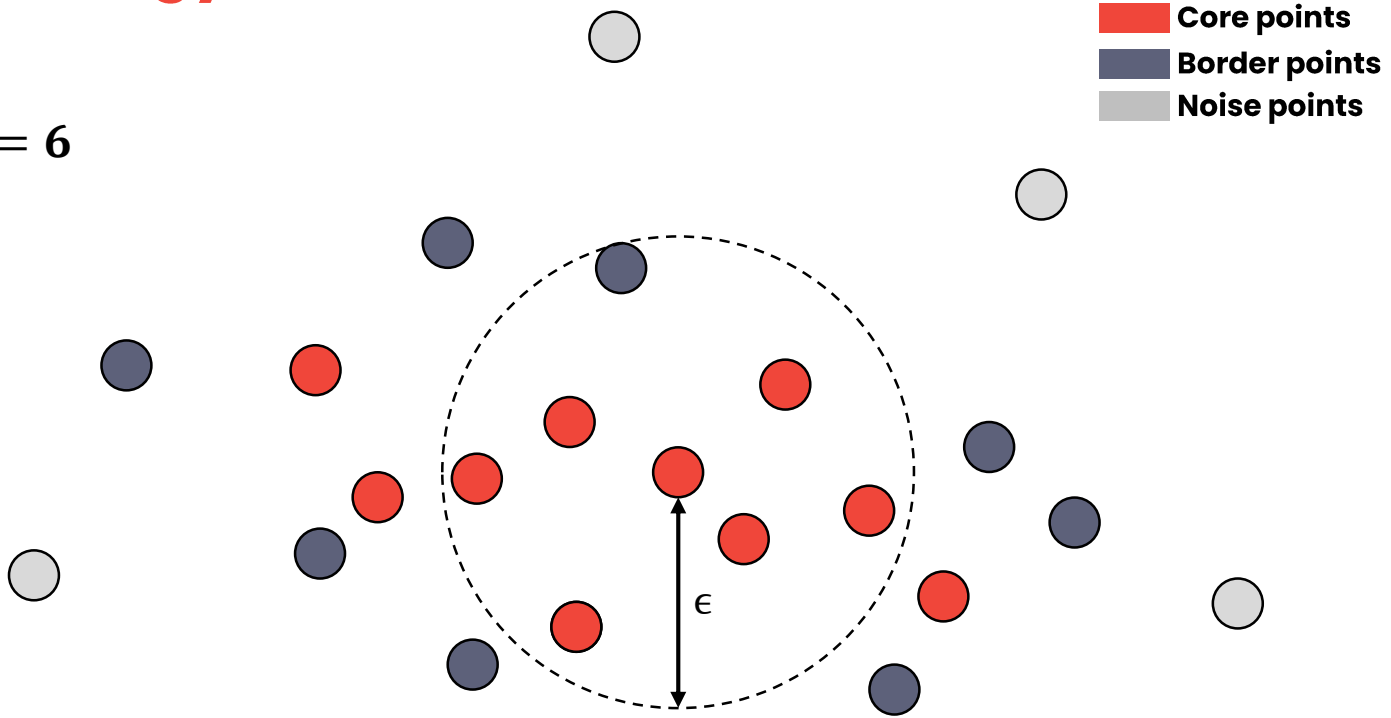
MinPts = 6



p and q are density-connected

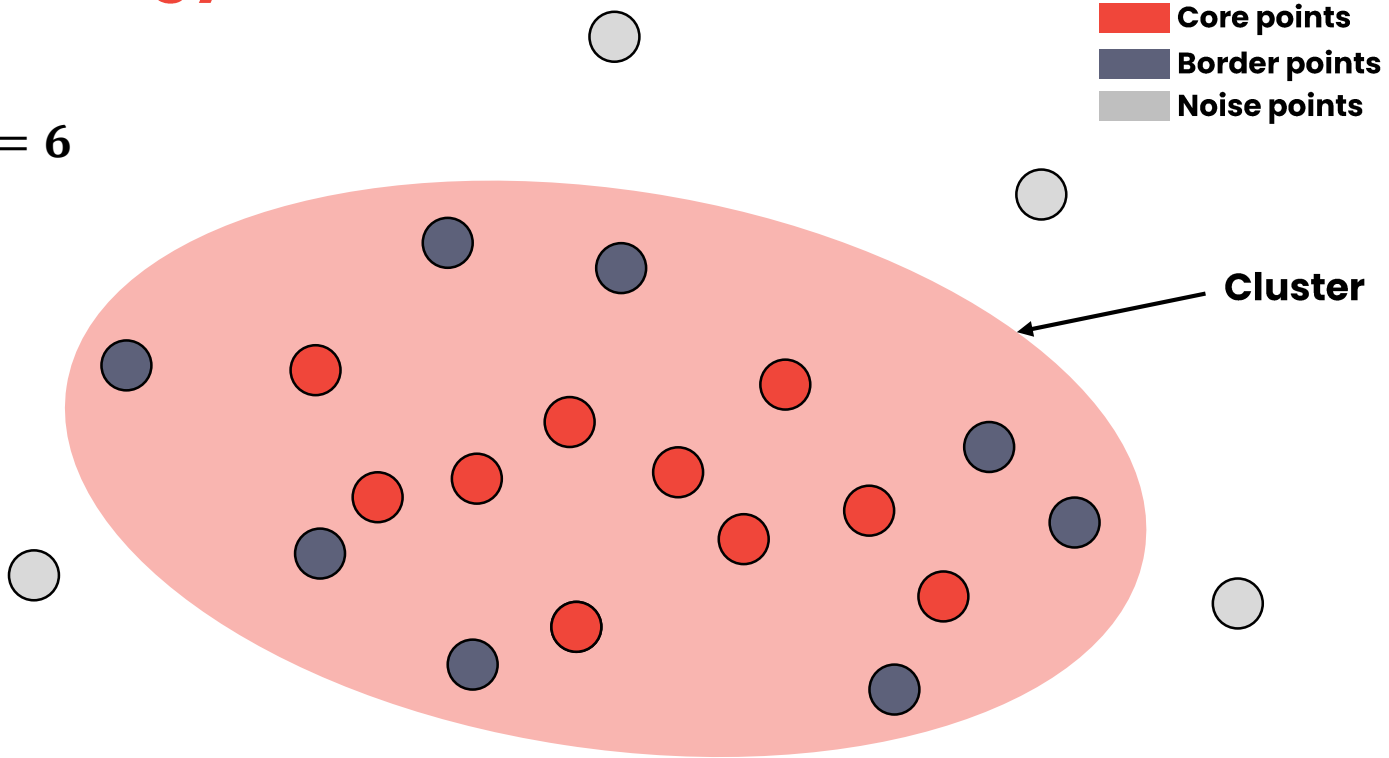
Terminology

MinPts = 6



Terminology

MinPts = 6



DBSCAN Algorithm

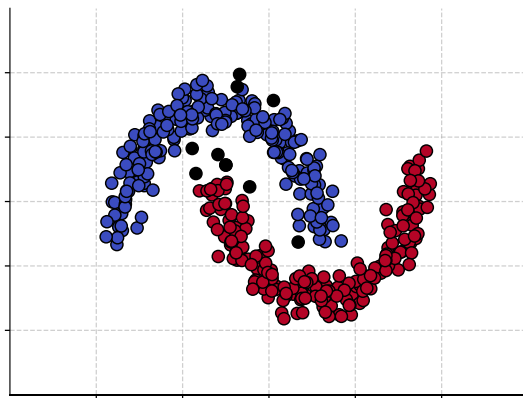
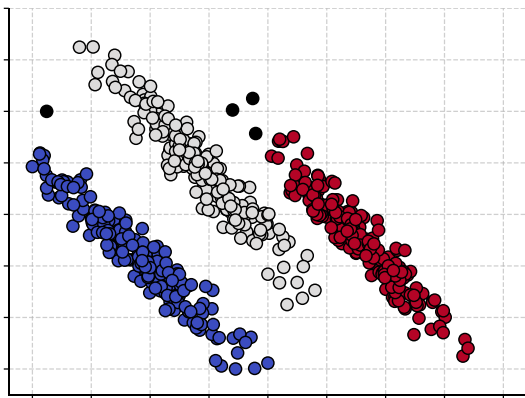
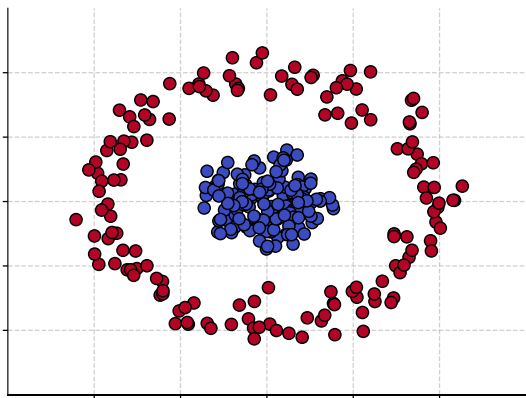
```
1  for every point  $p \in X$ :  
2      if  $p$  is a core point:  
3          label  $p$  with a unique cluster id  
4          for every point  $q \in X$  which is density-reachable from  $p$ :  
5              label  $q$  with the same cluster id as  $p$   
6      else if  $p$  has no label:  
7          label  $p$  as "noise" # might get relabeled later
```

More Detailed Pseudocode

```
1  def DBSCAN(X,  $\epsilon$ , minPts):
2      c = 0                                # cluster index
3      for p in X:
4          if p.label is not None:          # previously processed
5              continue
6          neighbors = find_neighborhood(p, X,  $\epsilon$ ) # find points in e-neighborhood
7          if len(neighbors) < minPts:
8              p.label = "noise"            # if not core point label as 'noise' (for now)
9              continue
10         c = c + 1                          # increment cluster
11         p.label = c                        # label first point in the new cluster
12         S = neighbors - {p}                # e-neighbors of p which we add to the cluster and try to expand
13         for q in S:
14             if q.label == "noise":
15                 q.label = c                # it was labeled as noise, but it is actually a border point
16             if q.label is not None:
17                 continue                  # either border point or in some other cluster
18             q.label = c                    # add the point to the cluster
19             neighbors = find_neighborhood(q, X,  $\epsilon$ ) # find e-neighborhood
20             if len(neighbors) >= minPts:    # check if also core point
21                 S = S U neighbors
```

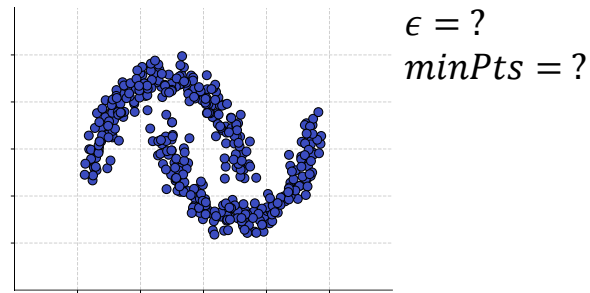
DBSCAN Results

- DBSCAN can handle *non-convex* cluster of various shapes.
- It doesn't require the *number of clusters* to be known in advance.
- The *distance metric* can also be considered a *hyperparameter*.
 - Euclidean distance is the most common.



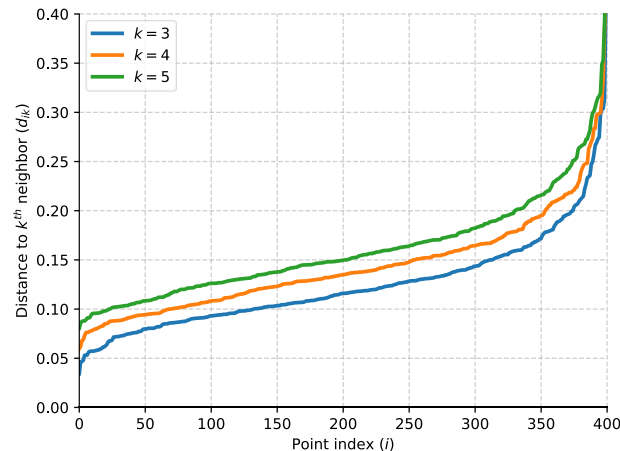
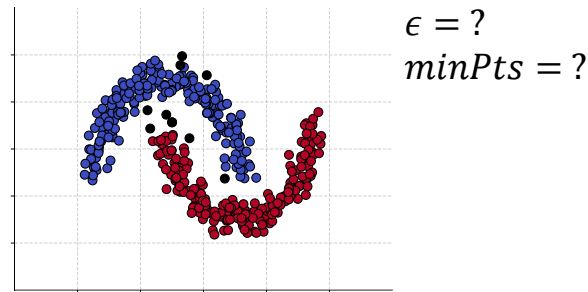
Choosing parameters

- DBSCAN is *very sensitive* to hyperparameters ϵ and $minPts$.



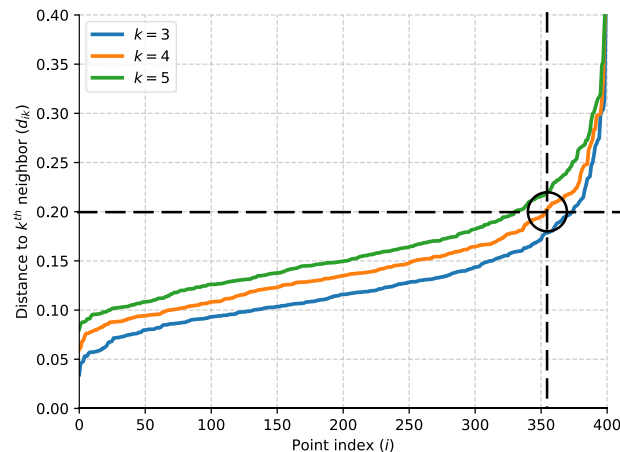
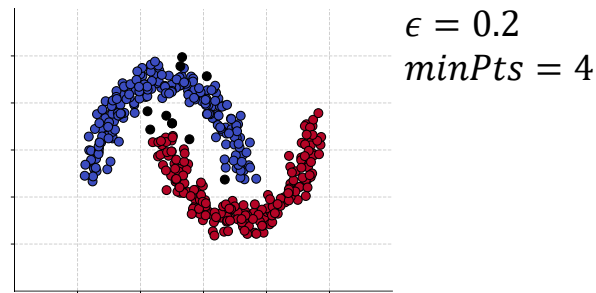
Choosing parameters

- DBSCAN is *very sensitive* to hyperparameters ϵ and $minPts$.
- An heuristic for choosing ϵ and $minPts$:
 - Choose a number k (typically ~ 4).
 - For each point $\vec{x}^{(i)}$, compute the distance d_{ik} to its k^{th} nearest neighbor.
 - Sort the points by d_{ik} and plot the corresponding curve.
 - Set $\epsilon \approx d_{ik}$ for a i for which the curve has a large change in slope.
 - Set $minPts = k$.
- All points under this threshold will be *core points*.
- It doesn't work very well for clusters with varying densities.



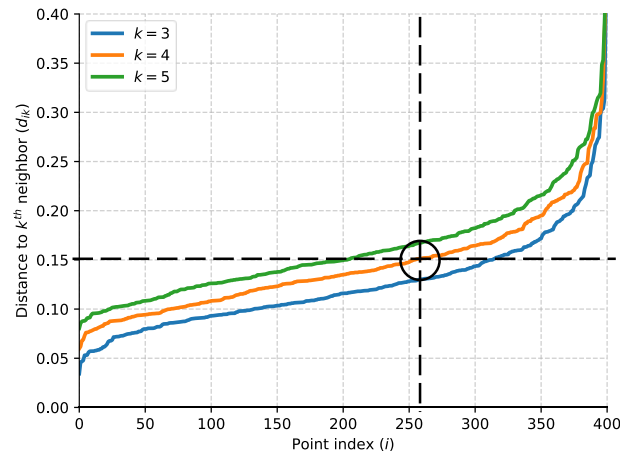
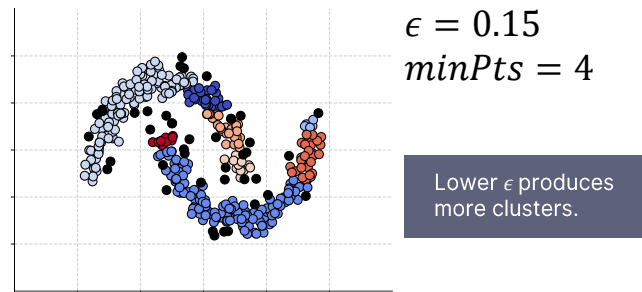
Choosing parameters

- DBSCAN is *very sensitive* to hyperparameters ϵ and $minPts$.
- An heuristic for choosing ϵ and $minPts$:
 - Choose a number k (typically ~ 4).
 - For each point $\vec{x}^{(i)}$, compute the distance d_{ik} to its k^{th} nearest neighbor.
 - Sort the points by d_{ik} and plot the corresponding curve.
 - Set $\epsilon \approx d_{ik}$ for a i for which the curve has a large change in slope.
 - Set $minPts = k$.
- All points under this threshold will be *core points*.
- It doesn't work very well for clusters with varying densities.



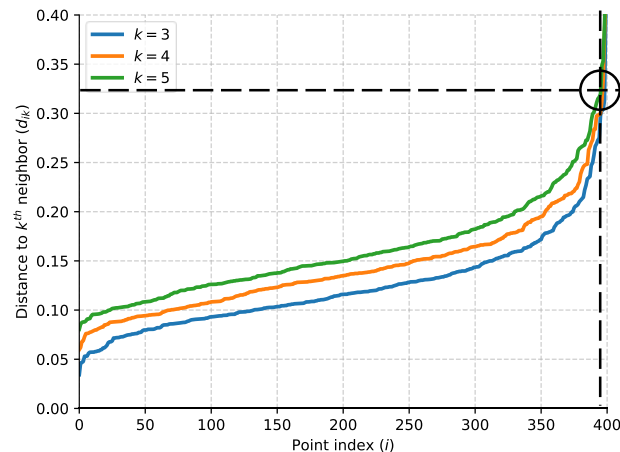
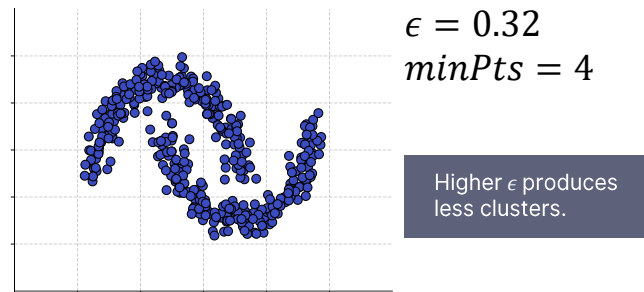
Choosing parameters

- DBSCAN is *very sensitive* to hyperparameters ϵ and $minPts$.
- An heuristic for choosing ϵ and $minPts$:
 - Choose a number k (typically ~ 4).
 - For each point $\vec{x}^{(i)}$, compute the distance d_{ik} to its k^{th} nearest neighbor.
 - Sort the points by d_{ik} and plot the corresponding curve.
 - Set $\epsilon \approx d_{ik}$ for a i for which the curve has a large change in slope.
 - Set $minPts = k$.
- All points under this threshold will be *core points*.
- It doesn't work very well for clusters with varying densities.



Choosing parameters

- DBSCAN is *very sensitive* to hyperparameters ϵ and $minPts$.
- An heuristic for choosing ϵ and $minPts$:
 - Choose a number k (typically ~ 4).
 - For each point $\vec{x}^{(i)}$, compute the distance d_{ik} to its k^{th} nearest neighbor.
 - Sort the points by d_{ik} and plot the corresponding curve.
 - Set $\epsilon \approx d_{ik}$ for a i for which the curve has a large change in slope.
 - Set $minPts = k$.
- All points under this threshold will be *core points*.
- It doesn't work very well for clusters with varying densities.



Summary

- **DBSCAN** is a *density-based* clustering algorithm which groups points which are closely packed together and marks as noise (outliers) points which are in low density regions.
- DBSCAN defines a cluster as a group of points which are **density-connected** to each other
 - Which means that for any pair of points, there is a chain of **core points** (i.e. points with enough neighbors in a given area) connecting them.
- Unlike *K-means*, DBSCAN can find clusters or *arbitrary shapes* and does not require the *number of clusters* to be known in advance.
- It has two *hyperparameters*, ϵ (the radius of neighborhood around core points) and *minPts* (the minimum number of neighbors which define a core point) which are quite hard to tune.

Keywords

DBSCAN

Density-based Clustering

Core-point

Border-point

ϵ -Neighborhood

Density-reachable

Density-connected