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.

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

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.

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

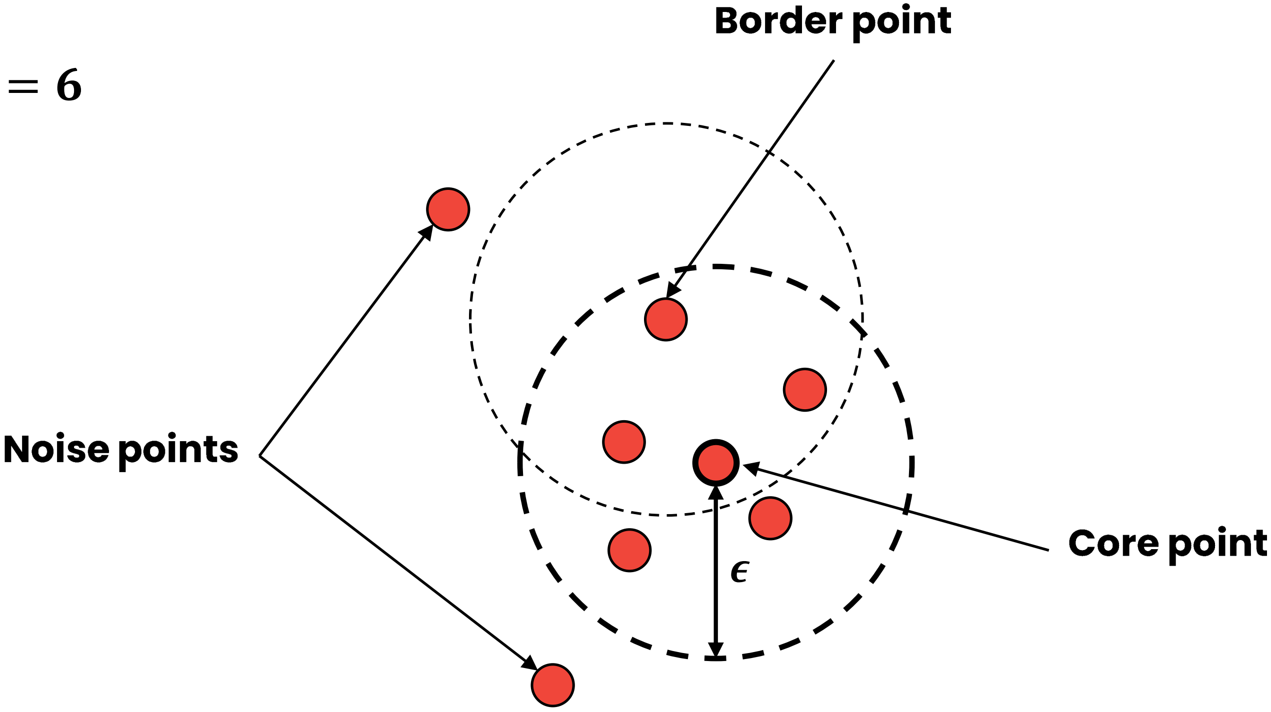
- A point p is a **core point** if its ϵ-neighborhood has at least $MinPts$ points.

$$|N_\epsilon(p)| \ge MinPts$$

- 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.

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

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.

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

- 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.

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

- q is **density-reachable** from p if there is a chain of points $p_1, p_2, \ldots, p_n \in X$ with $p_1 = p,\ p_n = q$ such that p_{i+1} is *directly density-reachable* from p_i

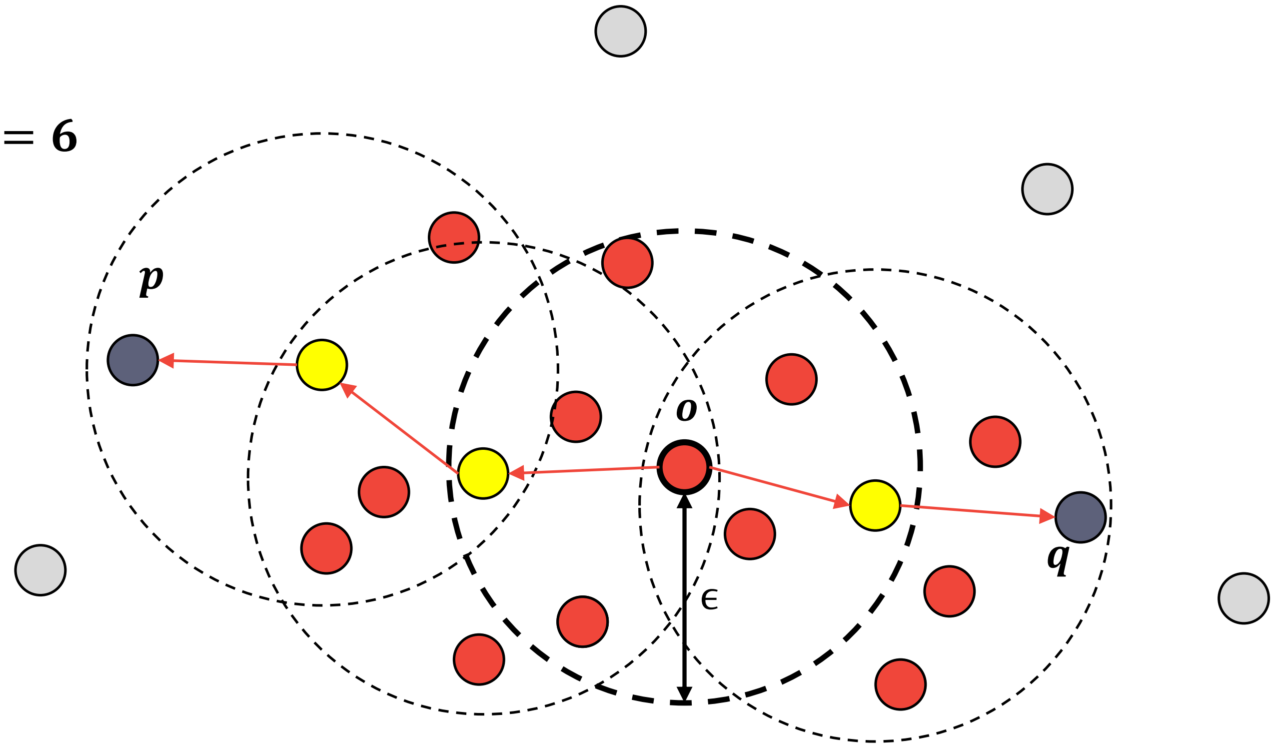
- p and q are **density-connected** if there is a point $o \in X$ such that both p and q are *density-reachable* from o.

- A **cluster** $C \subset X$ is a set of points which satisfies two conditions:
    - **Maximality:** $\forall p, q \in X, \text{if } p \in C$ and q is *density−reachable* from $p \Longrightarrow q \in C$
    - **Connectivity:** $\forall p, q \in C, \text{p}$ and q are *density−connected*

A cluster contains both core and border points.

Terminology

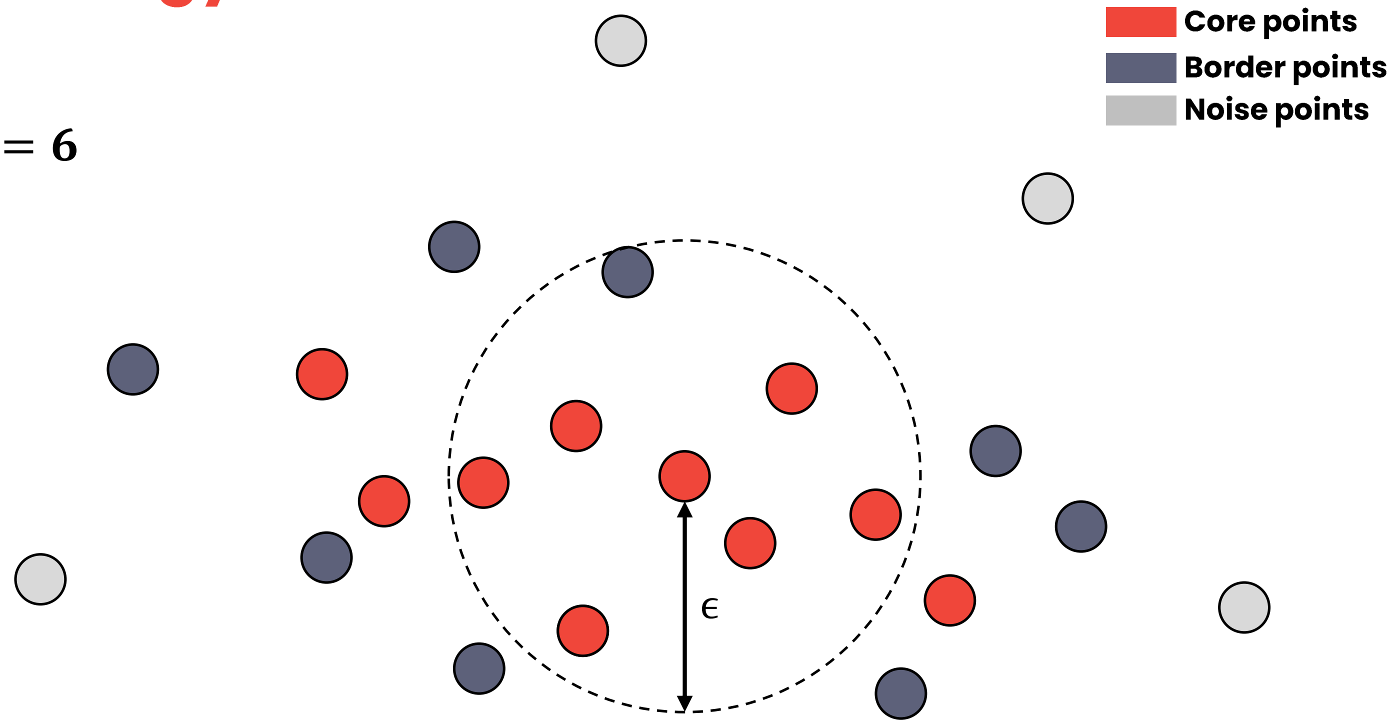
$MinPts = 6$



p and q are *density-connected*

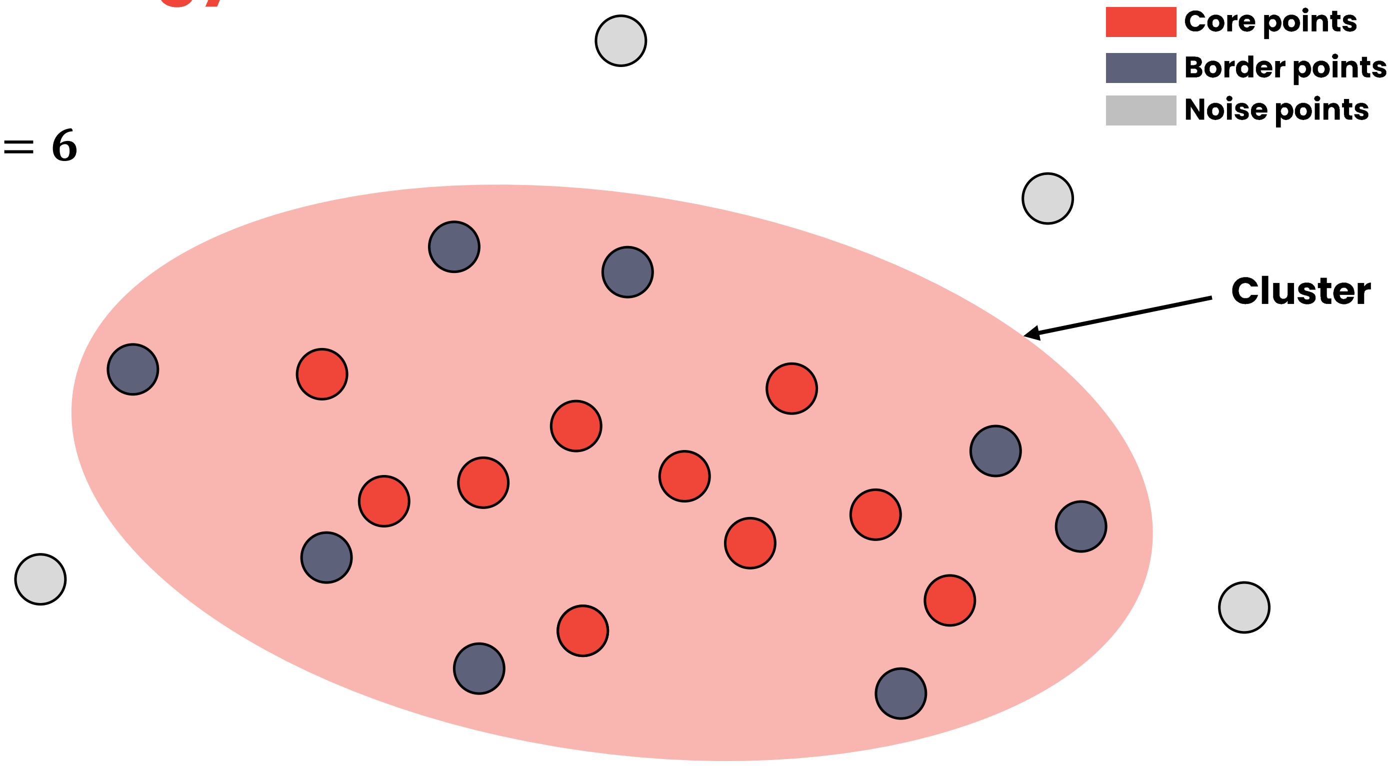
Terminology

$MinPts = 6$



Terminology

$MinPts = 6$



DBSCAN Algorithm

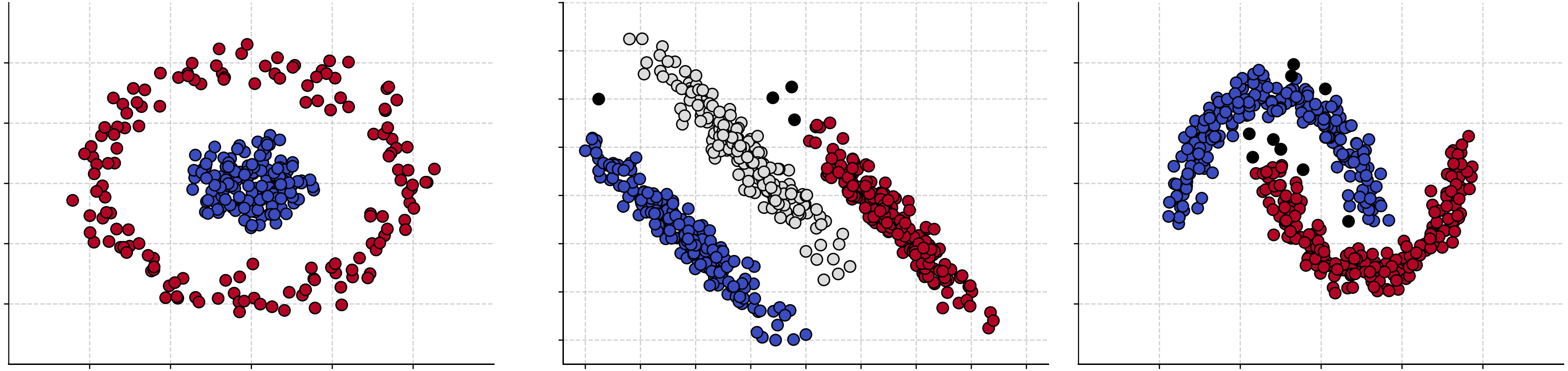
```
for every point p ∈ X:
    if p is a core point:
        label p with a unique cluster id
        for every point q ∈ X which is density-reachable from p:
            label q with the same cluster id as p
    else if p has no label:
        label p as "noise" # might get relabeled later
```

More Detailed Pseudocode

```
def DBSCAN(X, ϵ, minPts):
    c = 0                                               # cluster index
    for p in X:
        if p.label is not None:                          # previously processed
            continue
        neighbors = find_neighborhood(p, X, ϵ)           # find points in e-neighborhood
        if len(neighbors) < minPts:
            p.label = “noise”                             # if not core point label as ’noise’ (for now)
            continue
        c = c + 1                                        # increment cluster
        p.label = c                                      # label first point in the new cluster
        S = neighbors – {p}                          # e-neighbors of p which we add to the cluster and try to expand
        for q in S:
            if q.label == “noise”:
                q.label = c                             # it was labeled as noise, but it is actually a border point
            if q.label is not None:
                continue                                  # either border point or in some other cluster
            q.label = c                                   # add the point to the cluster
            neighbors = find_neighborhood(q, X, ϵ)        # find e-neighborhood
            if len(neighbors) >= minPts:                  # check if also core point
                S = S ∪ neighbors
```

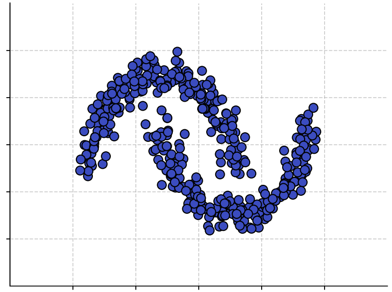
DBSCAN Results

- 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

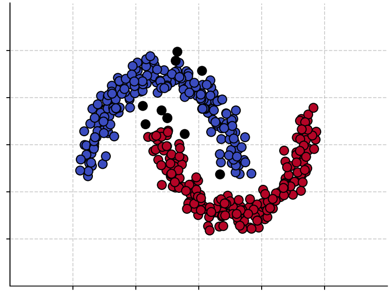
- DBSCAN is *very sensitive* to hyperparameters ϵ and *minPts*.



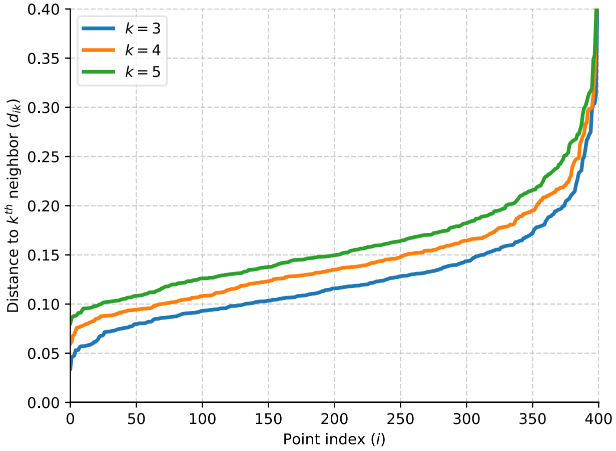
$\epsilon = ?$
$minPts = ?$

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.

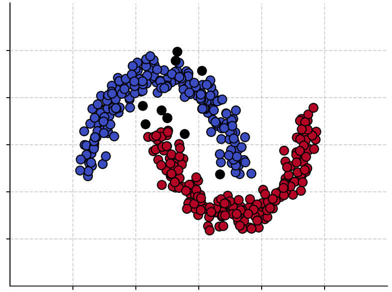


$\epsilon = ?$
$minPts = ?$

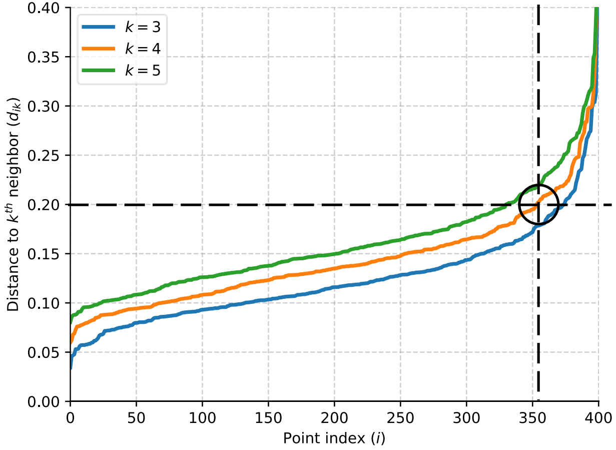


Choosing parameters

- 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.

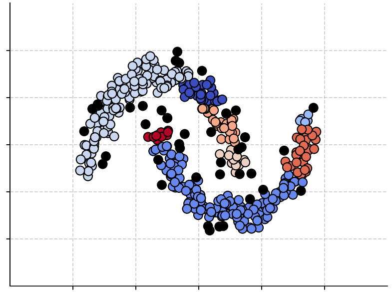


$\epsilon = 0.2$
$minPts = 4$



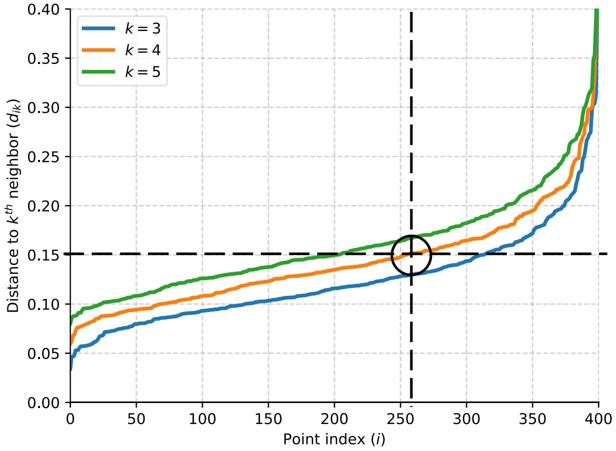
Choosing parameters

- 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.



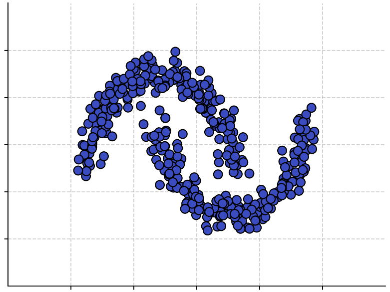
$\epsilon = 0.15$
$minPts = 4$

Lower ϵ produces more clusters.



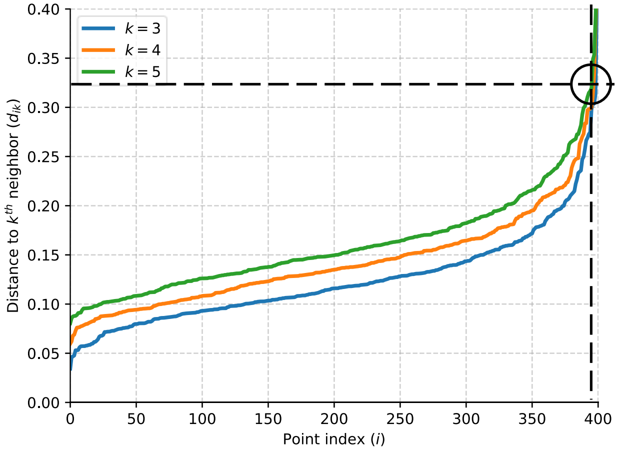
Choosing parameters

- 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.



$\epsilon = 0.32$
$minPts = 4$

Higher ϵ produces less clusters.



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