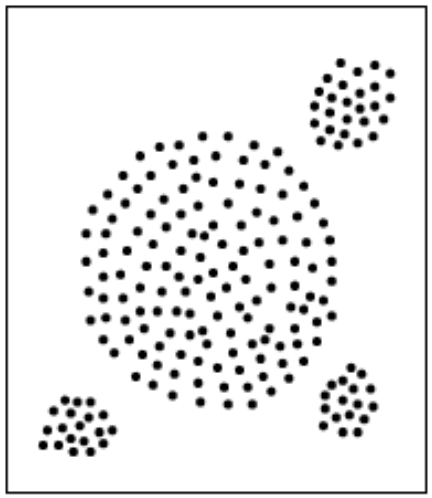
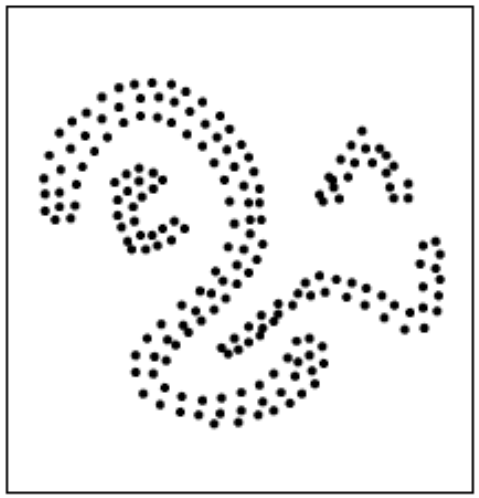
Motivation

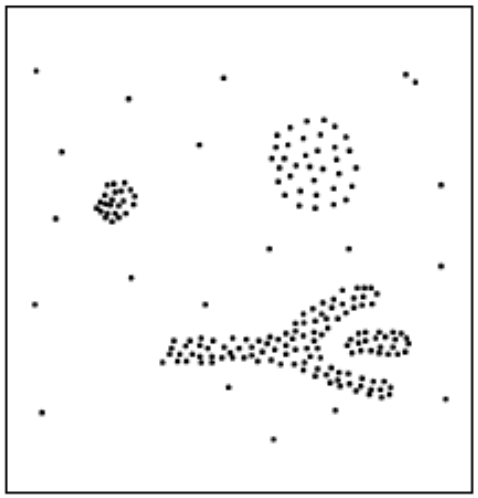
Idea: The clusters we see are higher density regions separated by lower density regions.



database 1



database 2



database 3

Motivation

- Dealing with clusters of different shapes and sizes.
- Dealing with large datasets efficiently.
- Unknown cluster count.
- Intuitive approach.
- Lets formalize this!

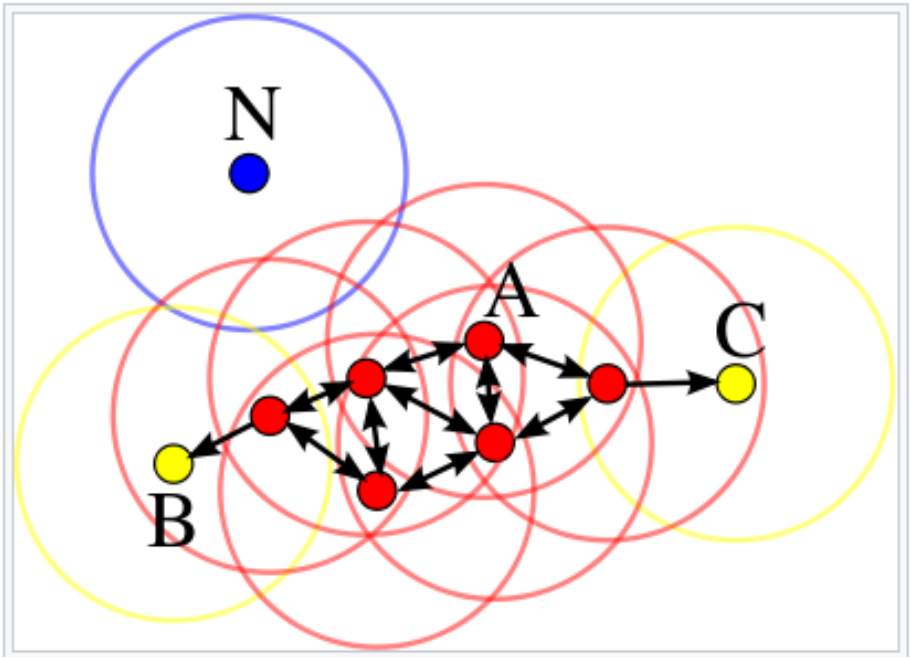
DBSCAN [1]

- Most popular density based clustering method
- Widely implemented, e.g., Python, Matlab, R etc.
- Published in 1996

DBSCAN Overview

3 point categories:

- Core Points (red)
- Border Points (yellow)
- Noise (blue)



DBSCAN Overview

2 parameters:

ϵ: Distance defining neighborhood

minPts: Minimum number of points to define a cluster

Point definitions:

Core Points: At least *minPts* points in ϵ-neighborhood.

Border Points: Within distance ϵ of core point.

Noise: All remaining points.

DBSCAN Algorithm

1. Set *minPts* and ϵ.
2. Choose random unvisited point.
3. If core point start new cluster otherwise classify as noise and choose another unvisited point.
4. Add all *reachable* core points to the cluster and their border points.
5. Repeat steps 2-4 until all points have been visited.

Notes on DBSCAN

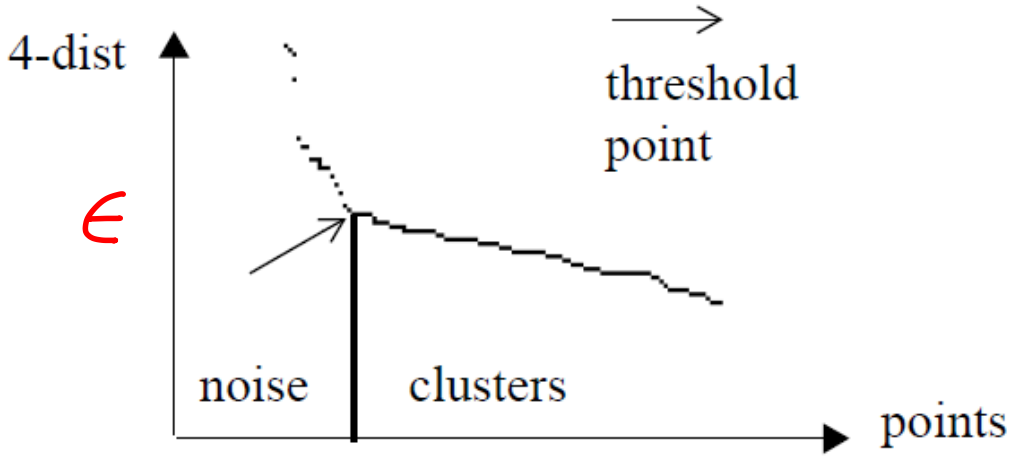
- Cluster count not a parameter.
- Any distance function can be used (usually euclidean).
- Includes noise category.
- Run DBSCAN interactively at:
  `https://tinyurl.com/Naftali-Harris`

Parameters of DBSCAN

- How do parameters ϵ and *minPts* affect results?
- $\uparrow \epsilon \implies \uparrow$ Reachability $\implies \downarrow$ Cluster Count
- $\uparrow$ *minPts* $\implies \downarrow$ Core point count $\implies \downarrow$ Reachability $\implies \uparrow$ Noise Point Count
- ϵ directly controls reachability.
- *minPts* controls line between border points and core points.

Setting parameters of DBSCAN

- Heuristic approach in original paper.
  k-**dist graph**:



- **General approach:** Grid search + cluster validation measure (e.g. Silhouette coefficient).
  - Makes sense for fast methods.

Pros and Cons of DBSCAN

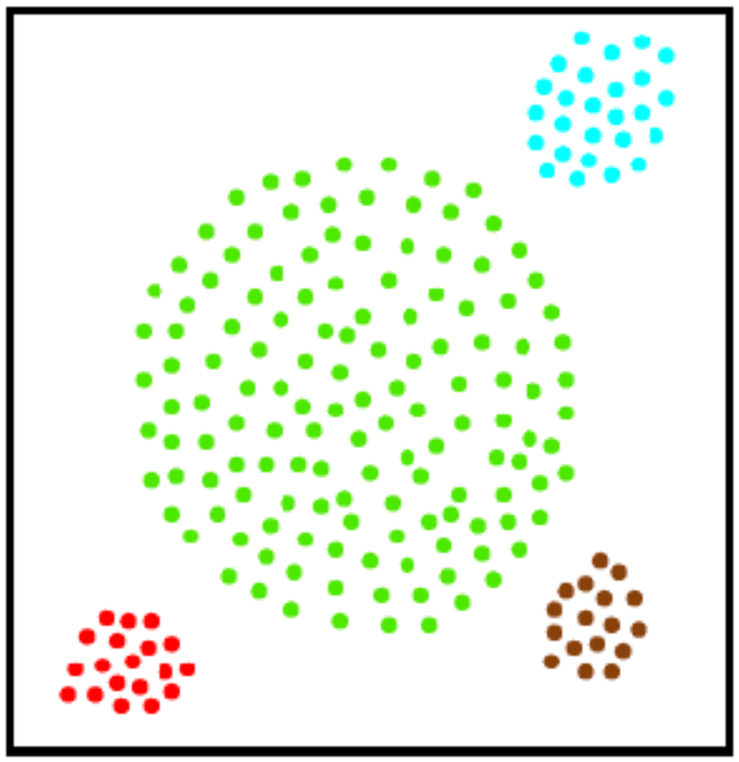
Pros

- Deals with arbitrarily shaped clusters.
- Built in noise/outlier detection.
- Only requires pairwise distances.

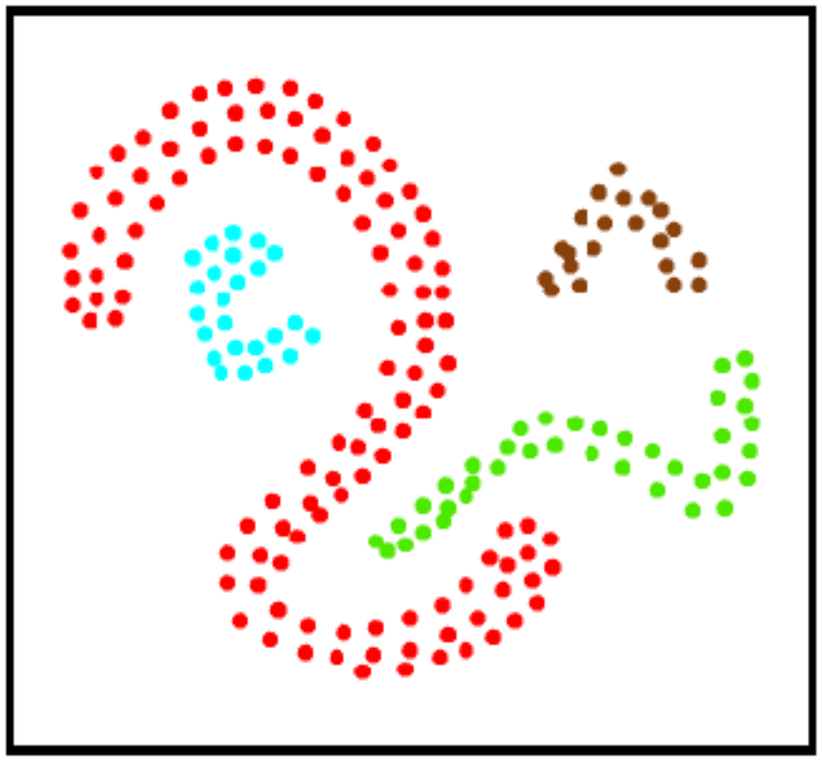
Cons

- Does not work well when clusters have very different densities (why?).

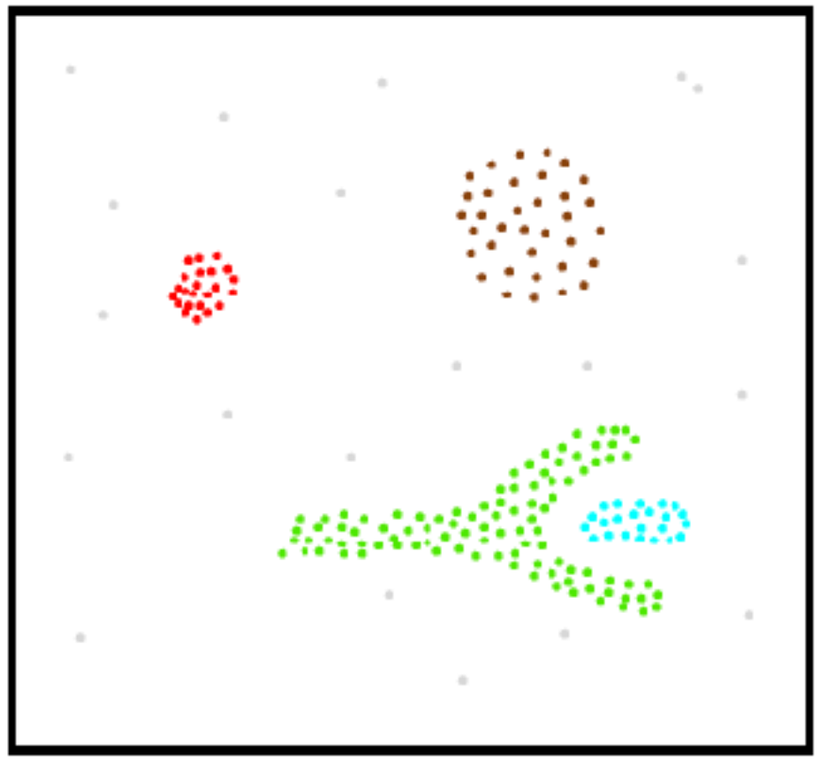
Examples of DBSCAN



database 1



database 2



database 3