Fundamentals of AI

Clustering

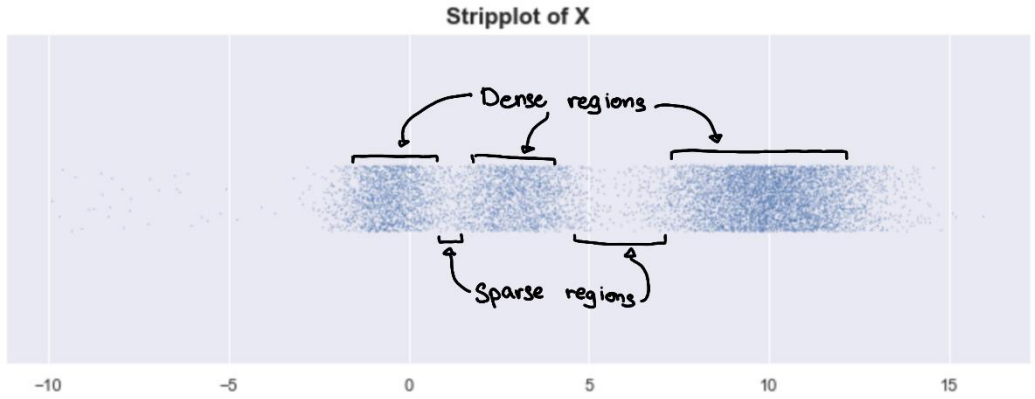
Density-based and graph-based clustering*

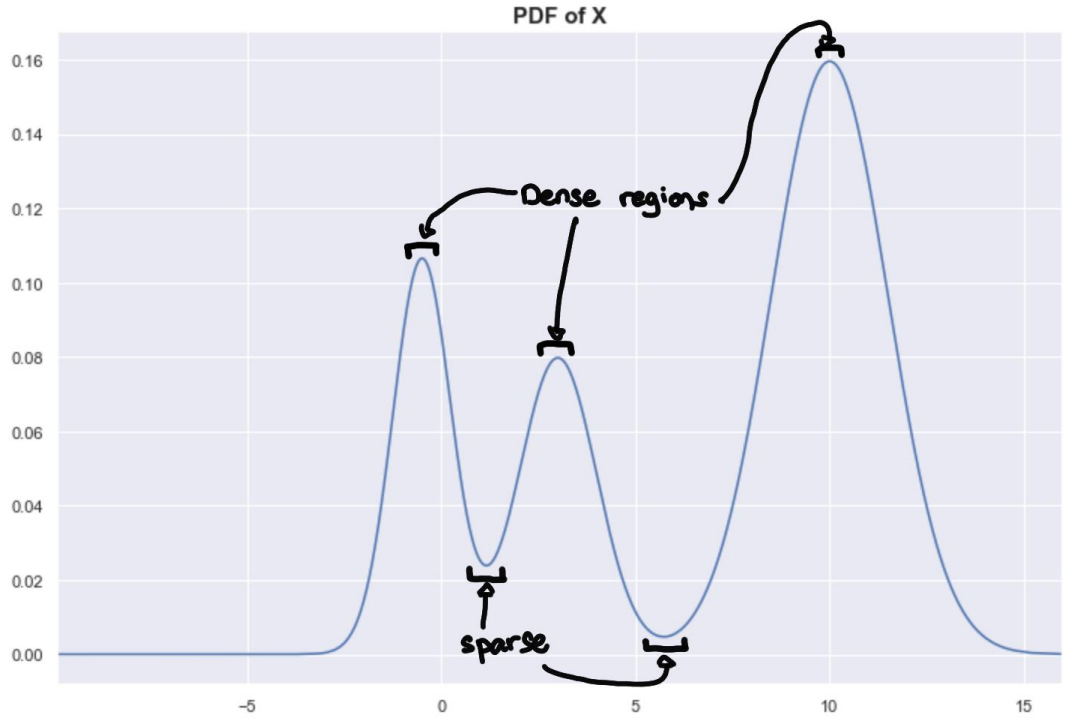
Some images in this lecture are used from: https://www.kdnuggets.com/2020/02/understanding-density-based-clustering.html

Distance-based clustering and its limitations

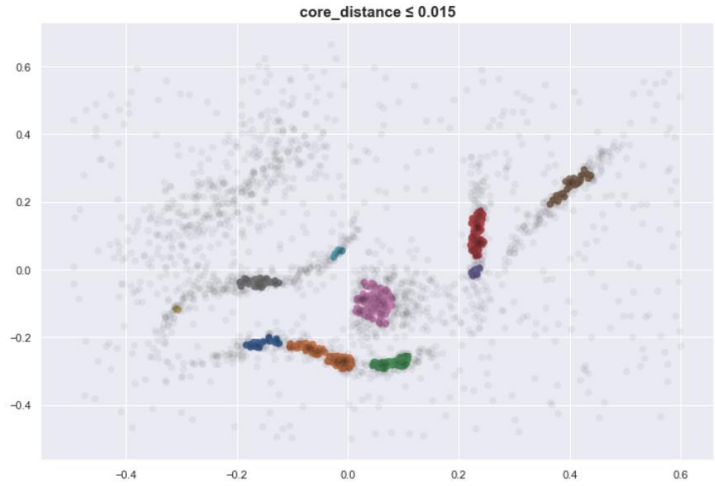
- Hard to find clusters with irregular shapes
- Hard to specify the number of clusters
- Some points are ‘in between’ clusters (outliers or background noise)

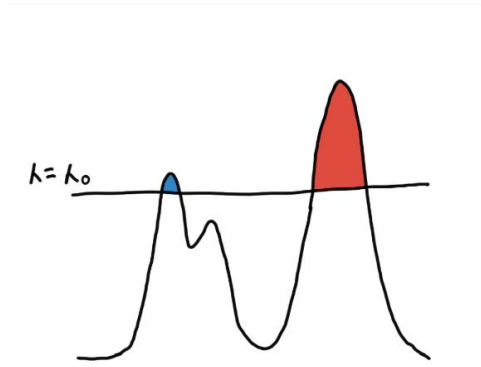
New concept: cluster as a probability density peak

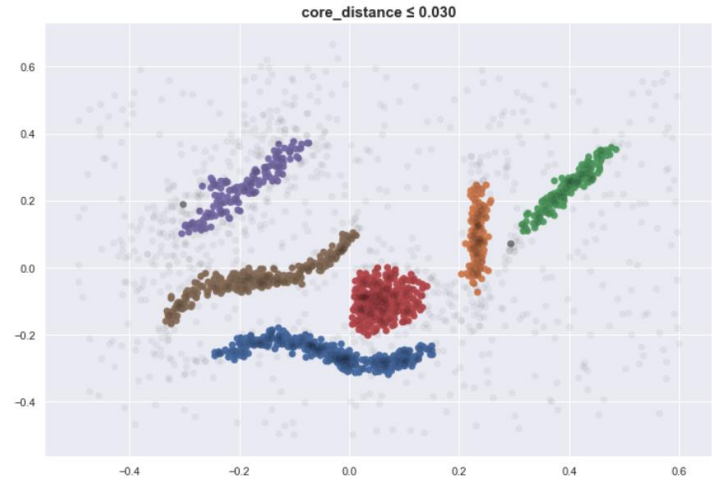


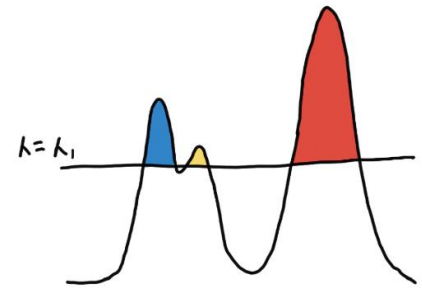


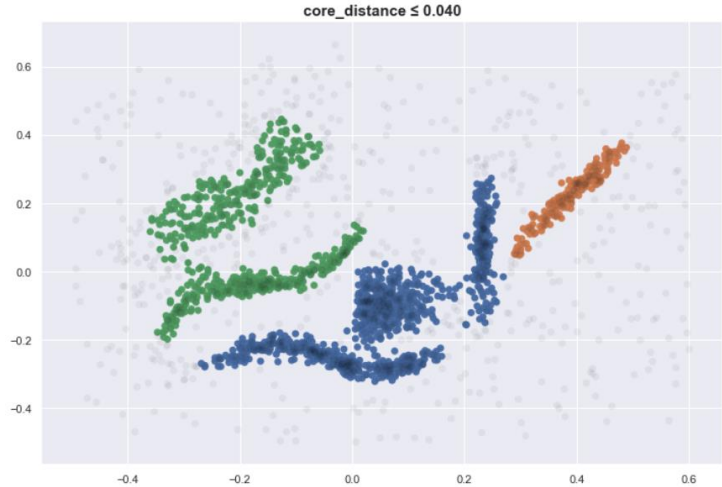
Cool, but how to define PDF in multi-dimensional space? Expensive and better to avoid at all

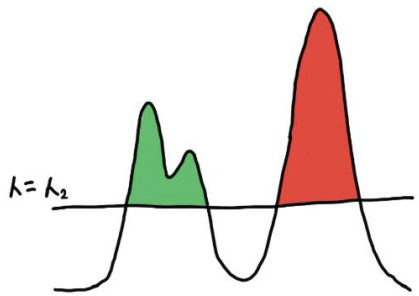

core_distance ≤ 0.015
0.6
0.4
0.2
0.0
-0.2
-0.4
-0.4
-0.2
0.0
0.2
0.4
0.6

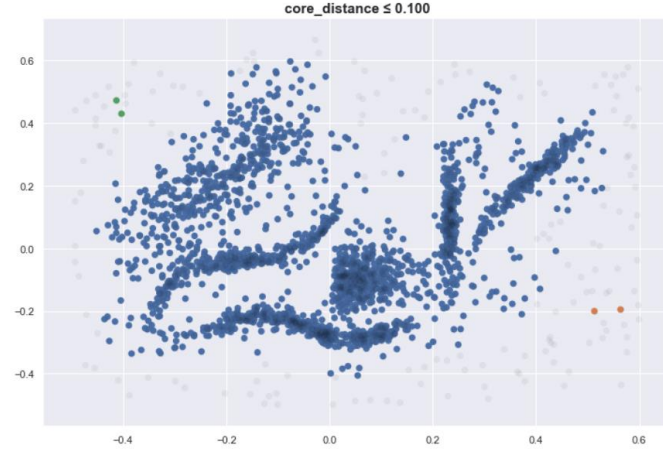

λ = λ0

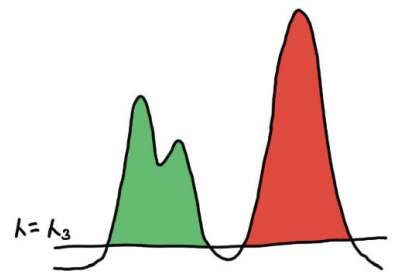

core_distance ≤ 0.030
0.6
0.4
0.2
0.0
-0.2
-0.4
-0.4
-0.2
0.0
0.2
0.4
0.6


λ = λ1

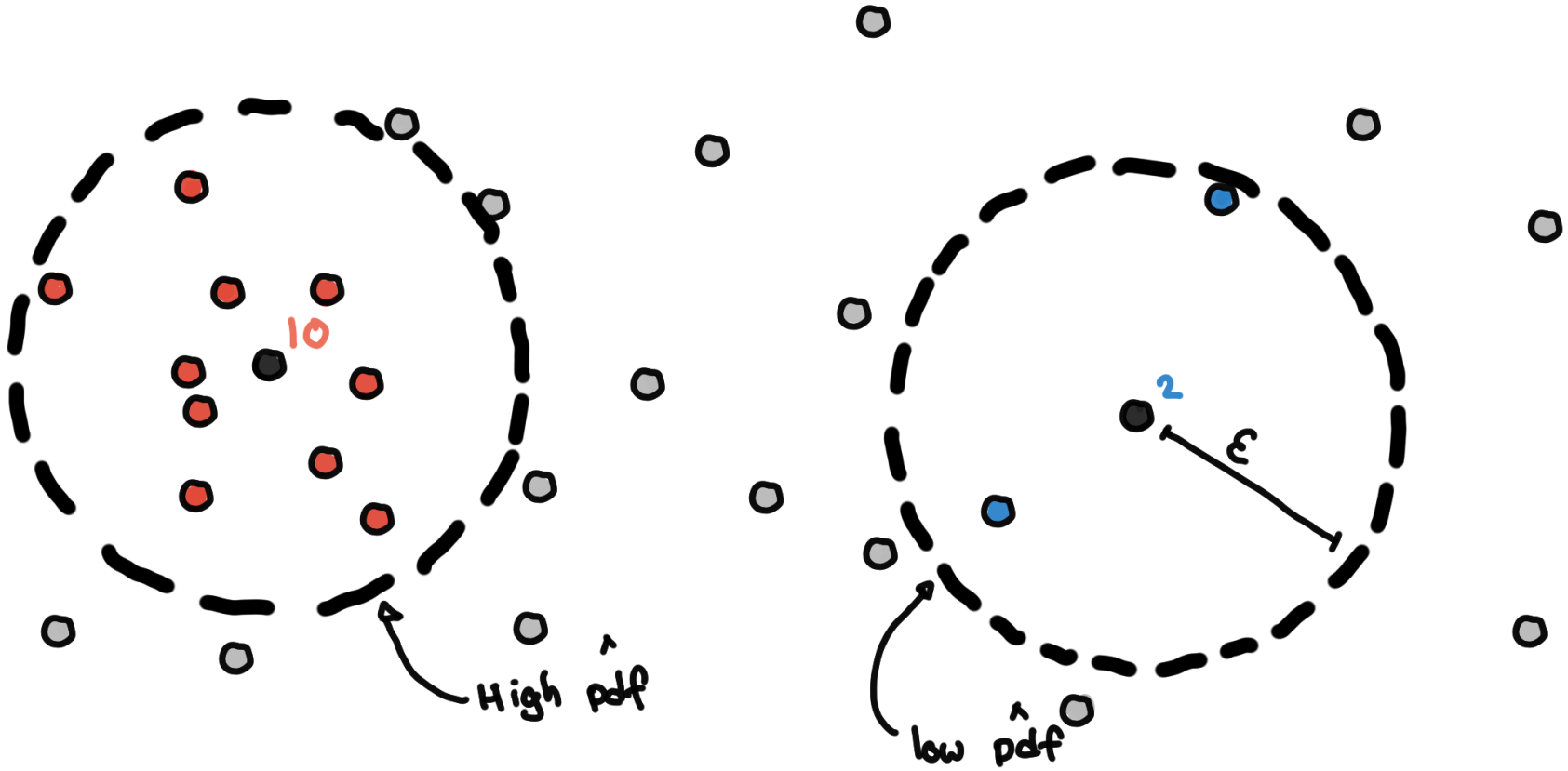

core_distance ≤ 0.040
0.6
0.4
0.2
0.0
-0.2
-0.4
-0.4
-0.2
0.0
0.2
0.4
0.6


λ = λ2


core_distance ≤ 0.100
0.6
0.4
0.2
0.0
-0.2
-0.4
-0.4
-0.2
0.0
0.2
0.4
0.6

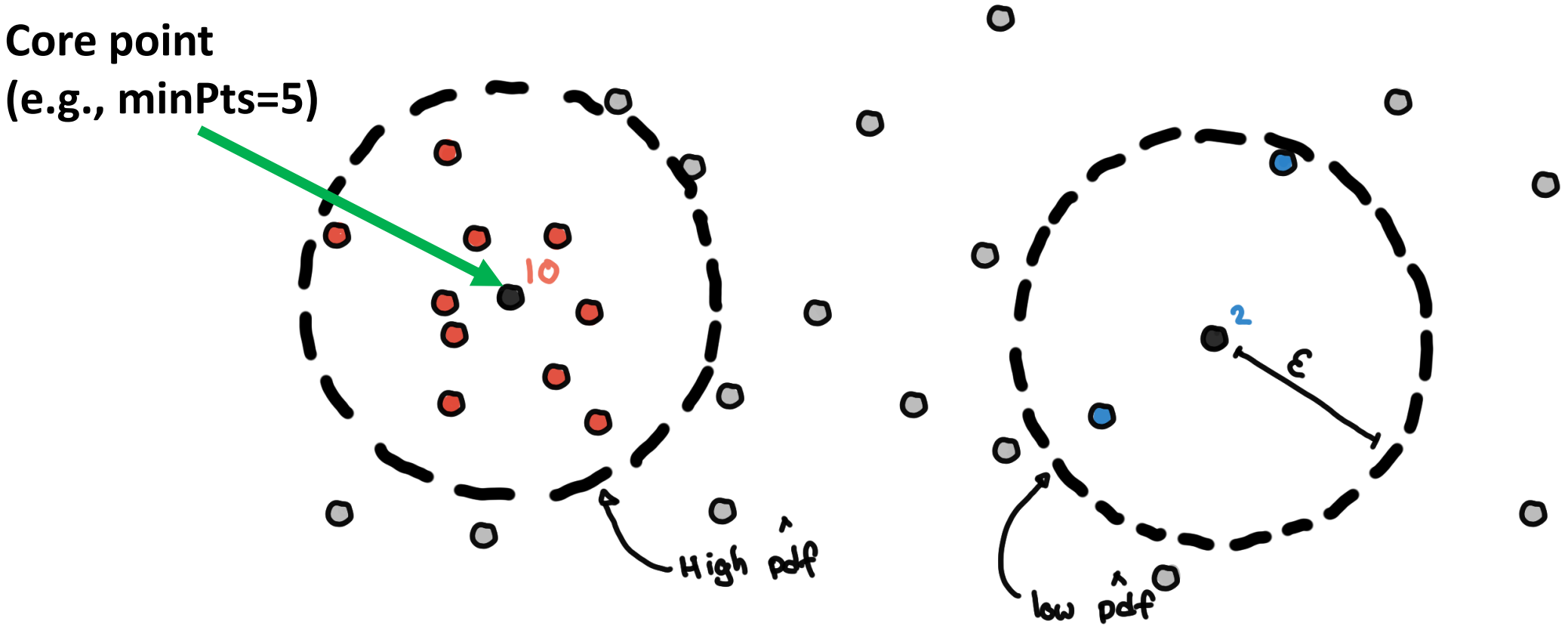

λ = λ3

Trick: count neighbours within ε-radius

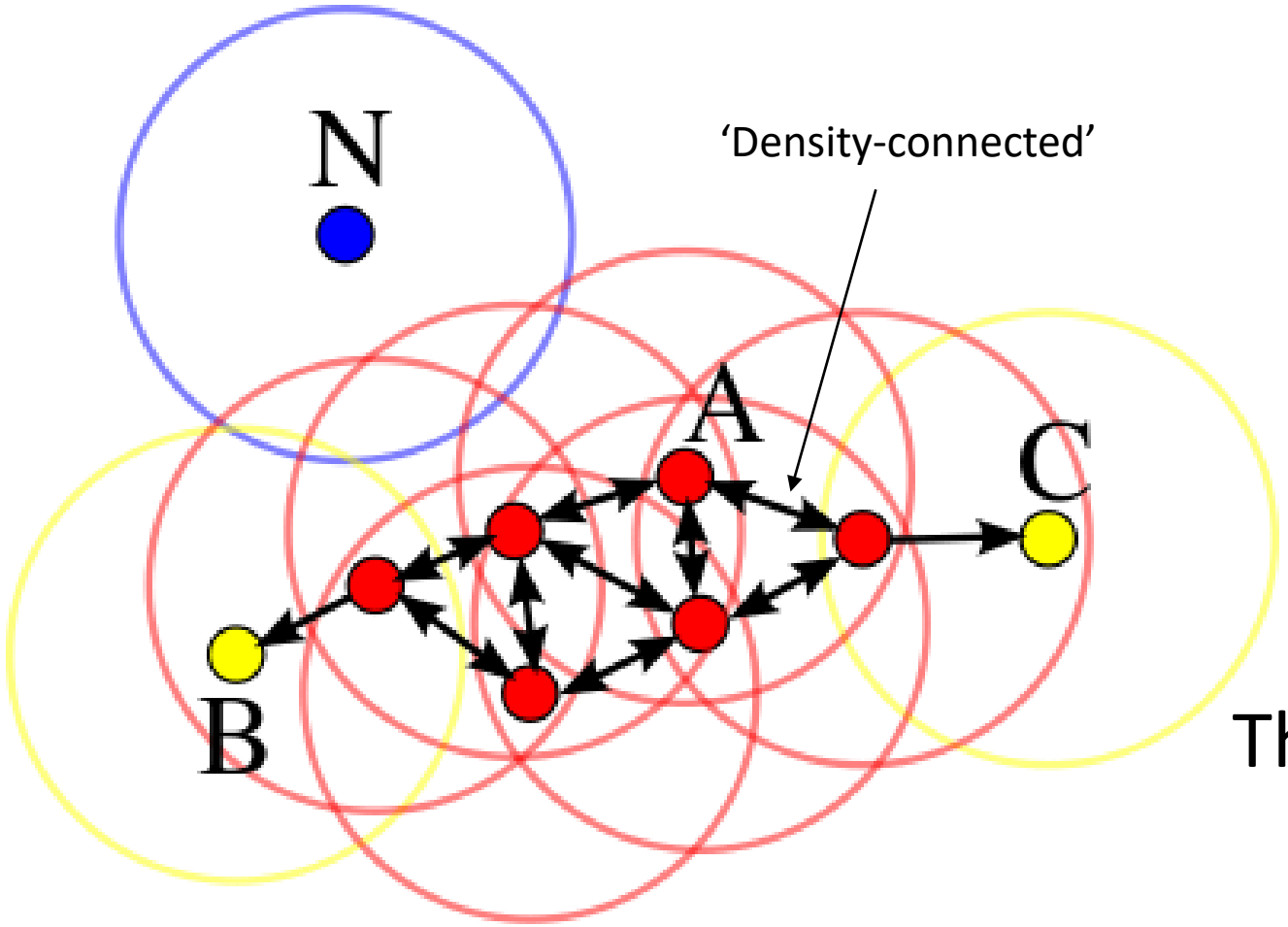


DBSCAN (Ester et al, 1996)

Parameters: ε and minPts



DBSCAN: graph of core points and density-reachable (peripherial) points



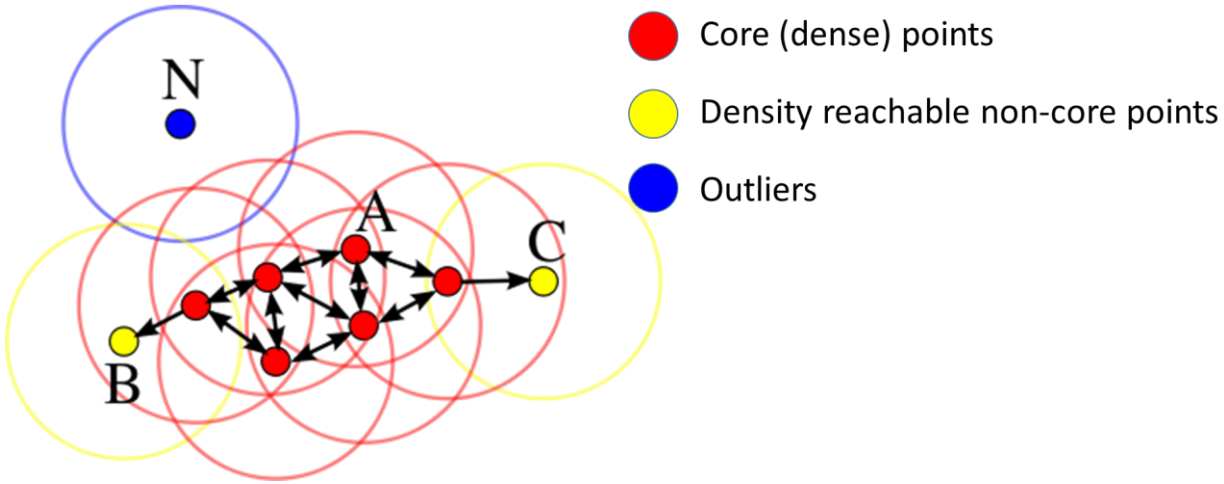
Core (dense) points

Density reachable points

Outliers

The graph is oriented!

DBSCAN: graph of core points and density-reachable (peripherial) points



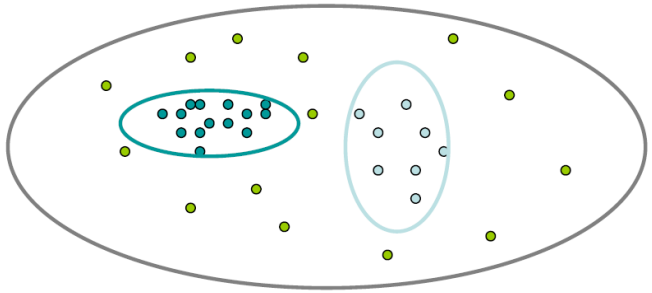
A cluster then satisfies two properties:

1. All points within the cluster are mutually density-connected.
2. If a point is density-reachable from some point of the cluster, it is part of the cluster as well.

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

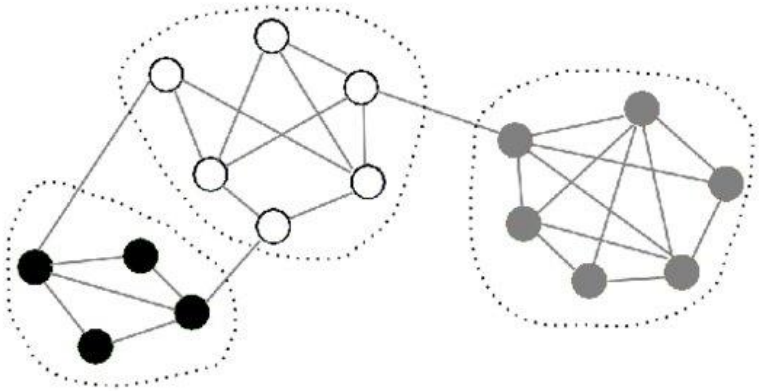
Comments on DBSCAN

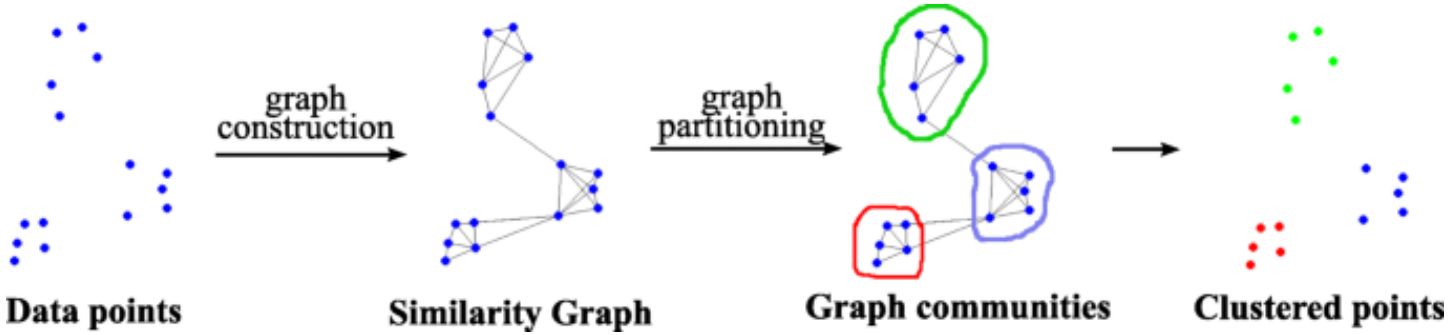


- Complexity is O(*n* log *n*)
- Unlike k-means and hierarchical, deal with the notion of noise
- Different clusters may have very different densities
- Very sensitive to the choice of ε
- Concentration of measures will spoil everything in high intrinsic dimensionalities
- Extensions: OPTICS, HDBSCAN, GDBSCAN
- Scikit learn implementation (arbitrary Lp metrics, accelerated neighbor search)

Graph-based clustering algorithms

- Cluster = tight community of the KNN graph





- The quality of communities is determined by **modularity**

Various definitions of the modularities

- E.g., **Louvain modularity**

$$Q = \frac{1}{2m} \sum_{ij} \left[A_{ij} - \frac{k_i k_j}{2m} \right] \delta(c_i, c_j),$$

In range [-0.5; 1]

where

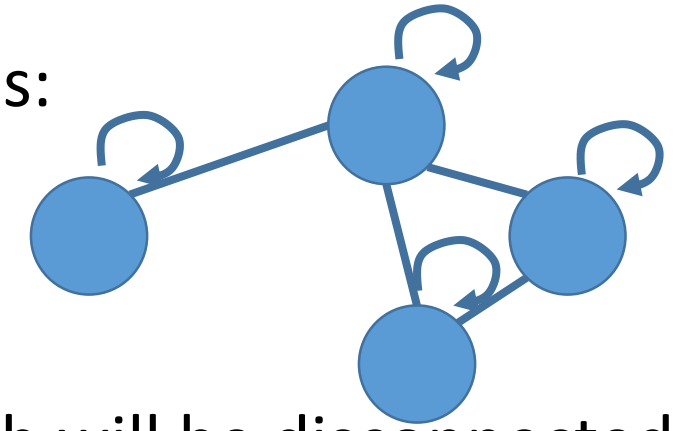
- A_{ij} represents the edge weight between nodes i and j;
- k_i and k_j are the sum of the weights of the edges attached to nodes i and j, respectively;
- m is the sum of all of the edge weights in the graph;
- c_i and c_j are the communities of the nodes; and
- δ is Kronecker delta function ($\delta(x, y) = 1$ if $x = y$, 0 otherwise).

Louvain clustering algorithm (a greedy one)

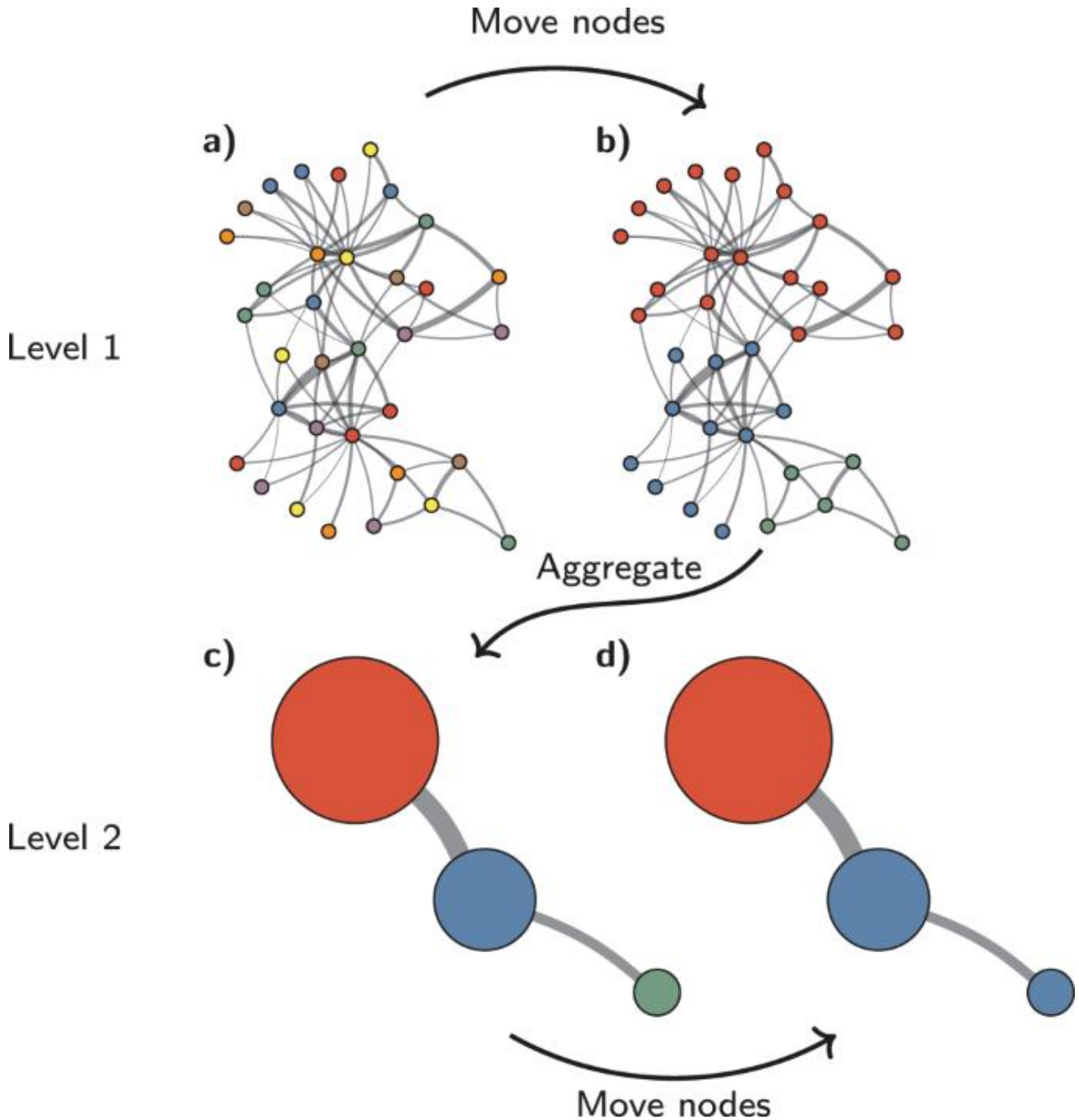
Base community search:

- At first, one node = one community
- We swap node *i* from its own community to the community of each of its neighbours
- For each such a swap, change in modularity is computed
- If no increase of modularity possible *i* remains in its own community

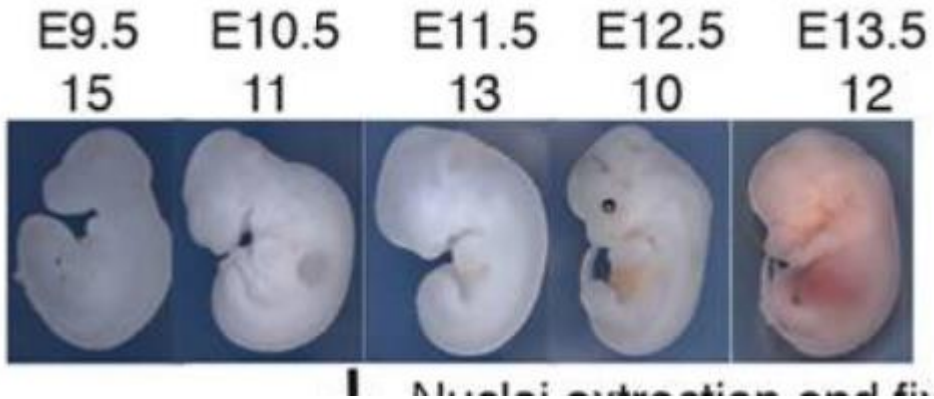
Finding communities in the graph of communities:



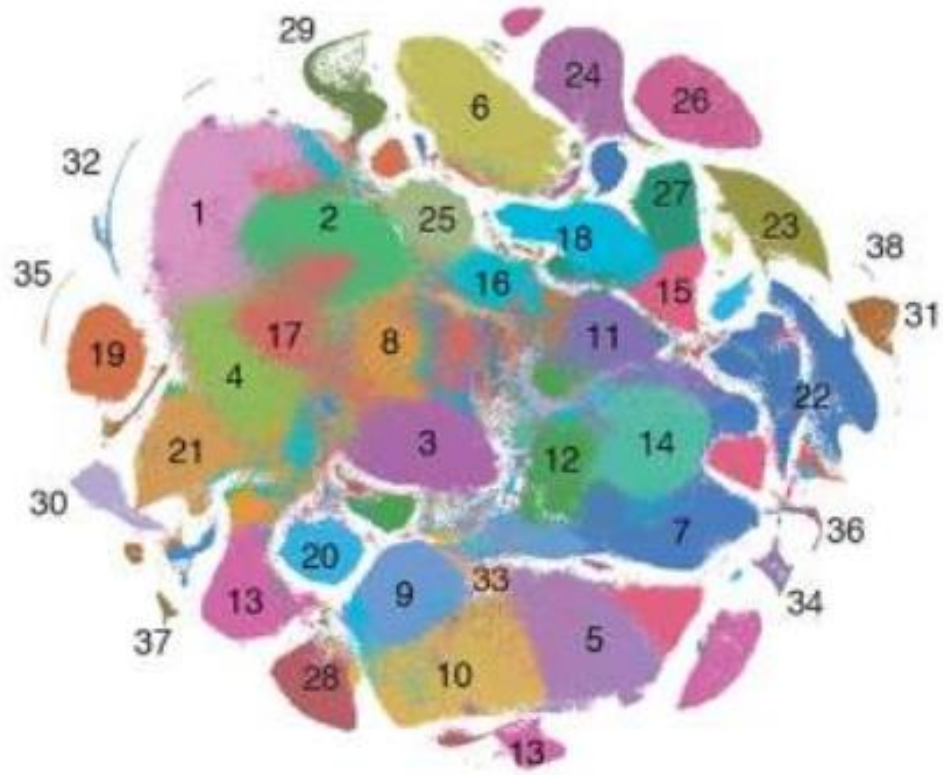
Cluster until one community remains or the graph will be disconnected


Move nodes
a)
b)
Level 1
Aggregate
c)
d)
Level 2
Move nodes

Graph-based clustering became new killer application in life sciences, replacing the hierarchical clustering



2 million data points – individual cells from mouse embryo



(from Cao et al, Nature, 2019)