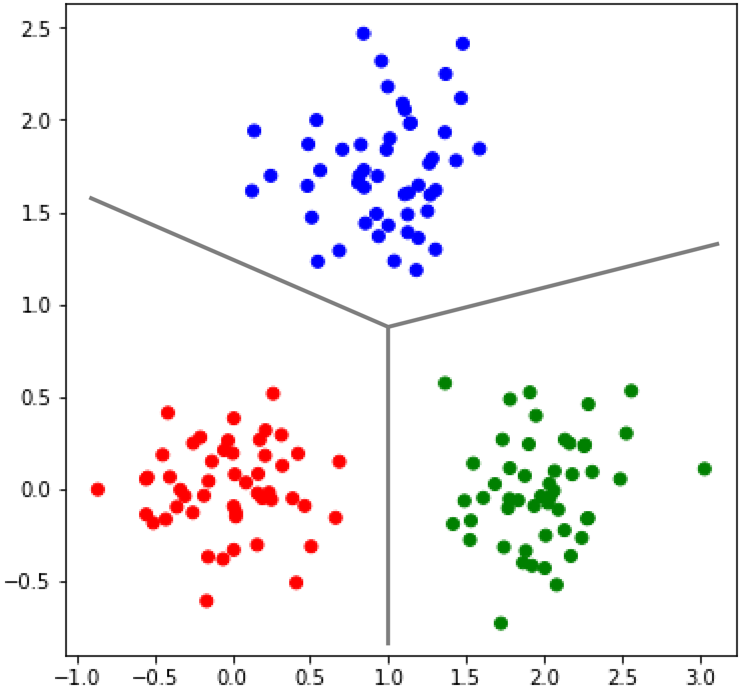
Clustering

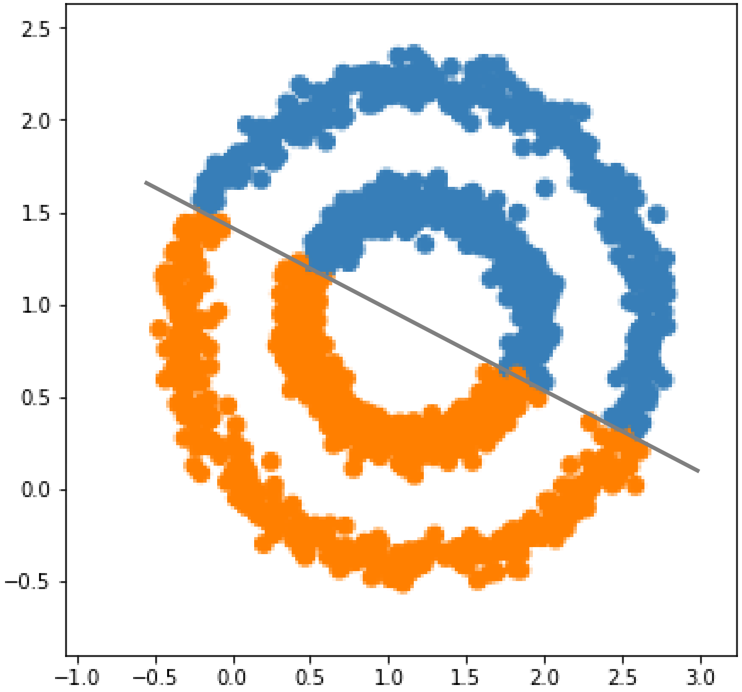
K-means, hierarchical clustering, DBSCAN

Reminder: limits of K-means

- K-means can be considered “linear” in that cluster boundaries are linear

(From the scikit-learn documentation)

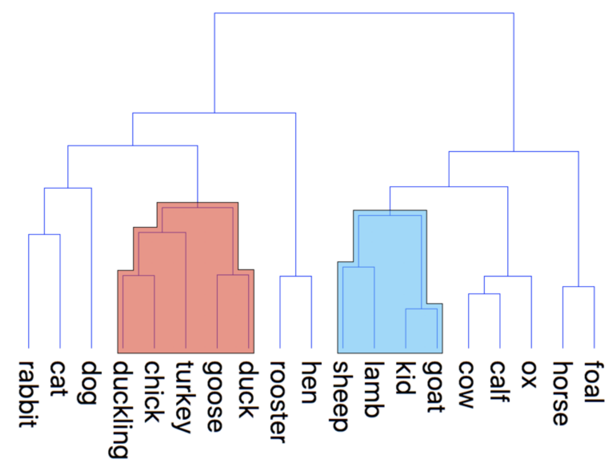




Hierarchical clustering

- This is not a specific clustering method but more like a family of methods
  - Advantage: it is not necessary to define the number of clusters *a priori* (but is still has to be selected at some point)
  - We simply need two components: an inter-cluster distance (or dissimilarity) and an intra-cluster distance
  - If these two distances are properly defined, we can work with any type of object (for instance strings, bits…)

(From Wikipedia)



Hierarchical clustering

- Assign each point to its own cluster:

$$\mathcal{C}_1 = \{\mathbf{x}_1\}, \ldots, \mathcal{C}_N = \{\mathbf{x}_N\}$$

- Find the two clusters closest to each other:

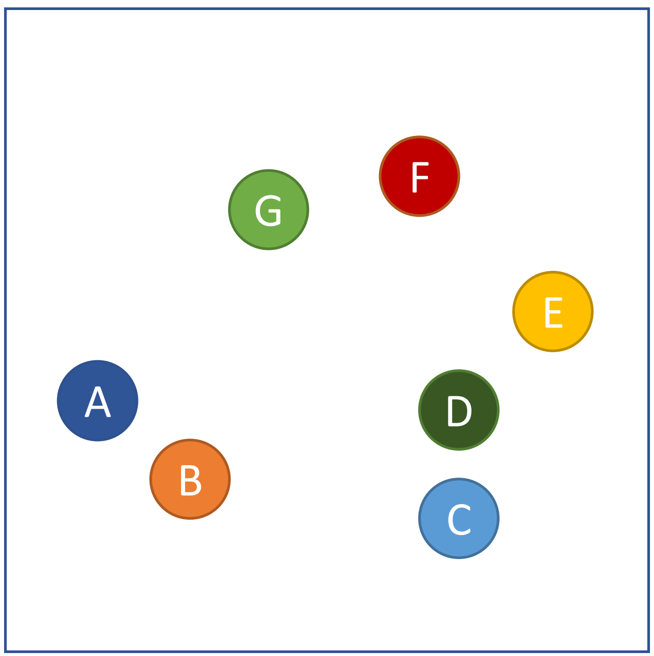
$$\mathcal{C}_i, \mathcal{C}_j = \underset{i,j}{\operatorname{argmin}} \; D(\mathcal{C}_i, \mathcal{C}_j)$$

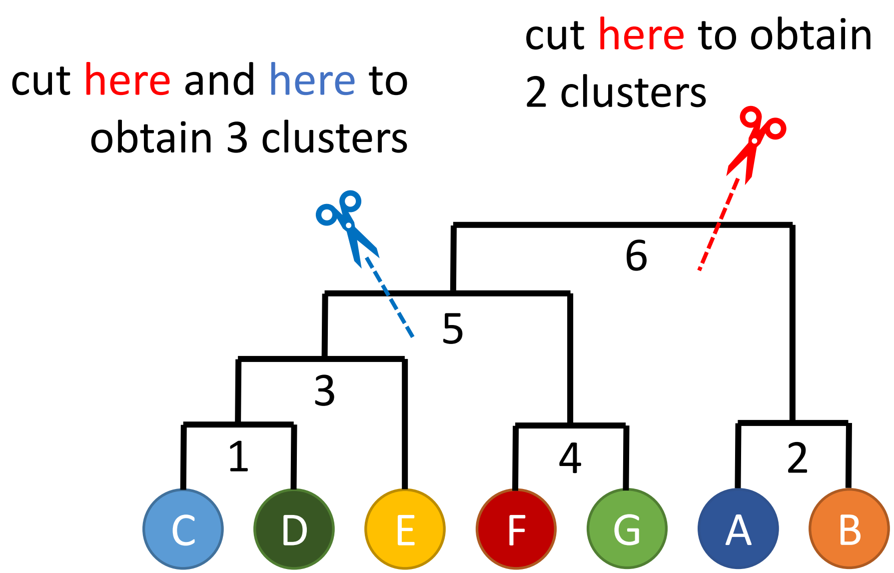
- Merge the two clusters, for instance
  - Update $\mathcal{C}_i = \mathcal{C}_i \cup \mathcal{C}_j$
  - Remove $\mathcal{C}_j$
  - (And keep track of these operations)

Repeat until there is only one cluster $\mathcal{C}_1$

Example

- Intra-cluster distance: $d(\mathbf{x}_i, \mathbf{x}_j) = \|\mathbf{x}_i - \mathbf{x}_j\|_2$
- Inter-cluster distance: $D(\mathcal{C}_i, \mathcal{C}_j) = \min_{\mathbf{x}_i \in C_i,\ \mathbf{x}_j \in C_j} d(\mathbf{x}_i, \mathbf{x}_j)$





Possible element-wise distances

- Euclidean or more generally Minkowski distance:

$$d(\mathbf{a}, \mathbf{b}) = \sqrt[p]{\sum_i (a_i - b_j)^p}$$

- Mahalanobis distance:

$$d(\mathbf{a}, \mathbf{b}) = \sqrt{(\mathbf{a} - \boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1} (\mathbf{b} - \boldsymbol{\mu})}$$

- Hamming distance (on bits)
  - `Hamming(`*`0100101, 1100100`*`) = 2`
- Levenshtein distance (on strings)
  - `Levenshtein(`*`levenshtein, levanstein`*`) = 2`
- …

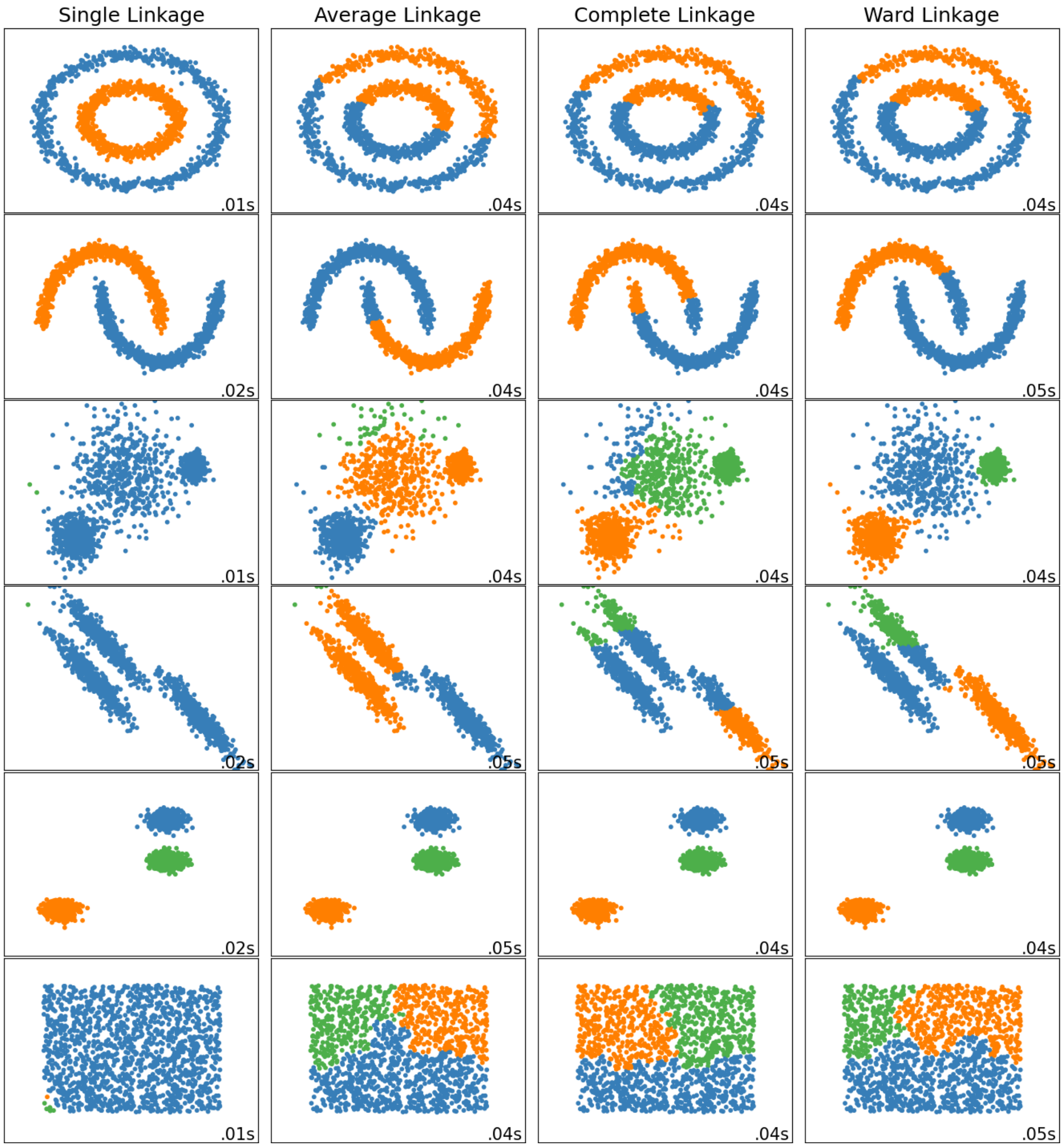
Possible group-wise distances

- Complete linkage: $D(\mathcal{C}_i, \mathcal{C}_j) = \max_{\mathbf{x}_i \in C_i,\ \mathbf{x}_j \in C_j} d(\mathbf{x}_i, \mathbf{x}_j)$
- Single linkage: $D(\mathcal{C}_i, \mathcal{C}_j) = \min_{\mathbf{x}_i \in C_i,\ \mathbf{x}_j \in C_j} d(\mathbf{x}_i, \mathbf{x}_j)$
- Average linkage: $\frac{1}{|\mathcal{C}_i| \cdot |\mathcal{C}_j|} \sum_{\mathbf{x}_i \in \mathcal{C}_i} \sum_{\mathbf{x}_j \in \mathcal{C}_j} d(\mathbf{x}_i, \mathbf{x}_j)$
- …

Distances in general

- Desirable properties of distances / dissimilarities:
  - Symmetry: $d(\mathbf{a}, \mathbf{b}) = d(\mathbf{b}, \mathbf{a})$
  - Separability: $d(\mathbf{a}, \mathbf{b}) = 0$ iff $\mathbf{a} = \mathbf{b}$
  - Triangular inequality: $d(\mathbf{a}, \mathbf{c}) \leq d(\mathbf{a}, \mathbf{b}) + d(\mathbf{b}, \mathbf{c})$
- Some methods may still work if this is not the case, but you may get unwanted results

Illustration

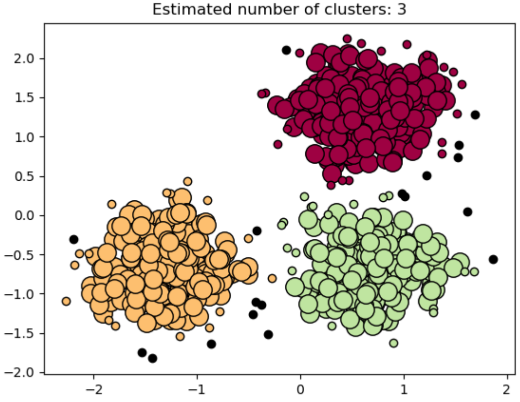


(From the scikit-learn documentation)

DBSCAN

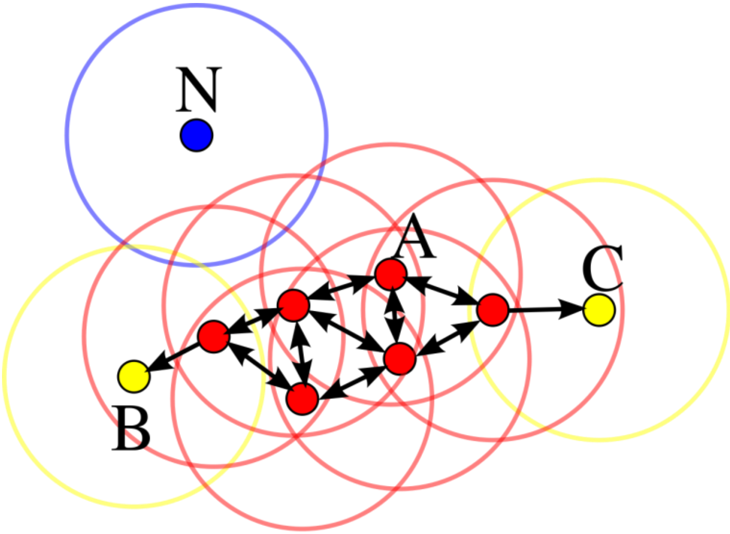
- Density Based Spatial Clustering of Applications with Noise (DBSCAN)

- Can automatically determine the number of clusters
- Can exclude some points from all clusters (“outliers”)



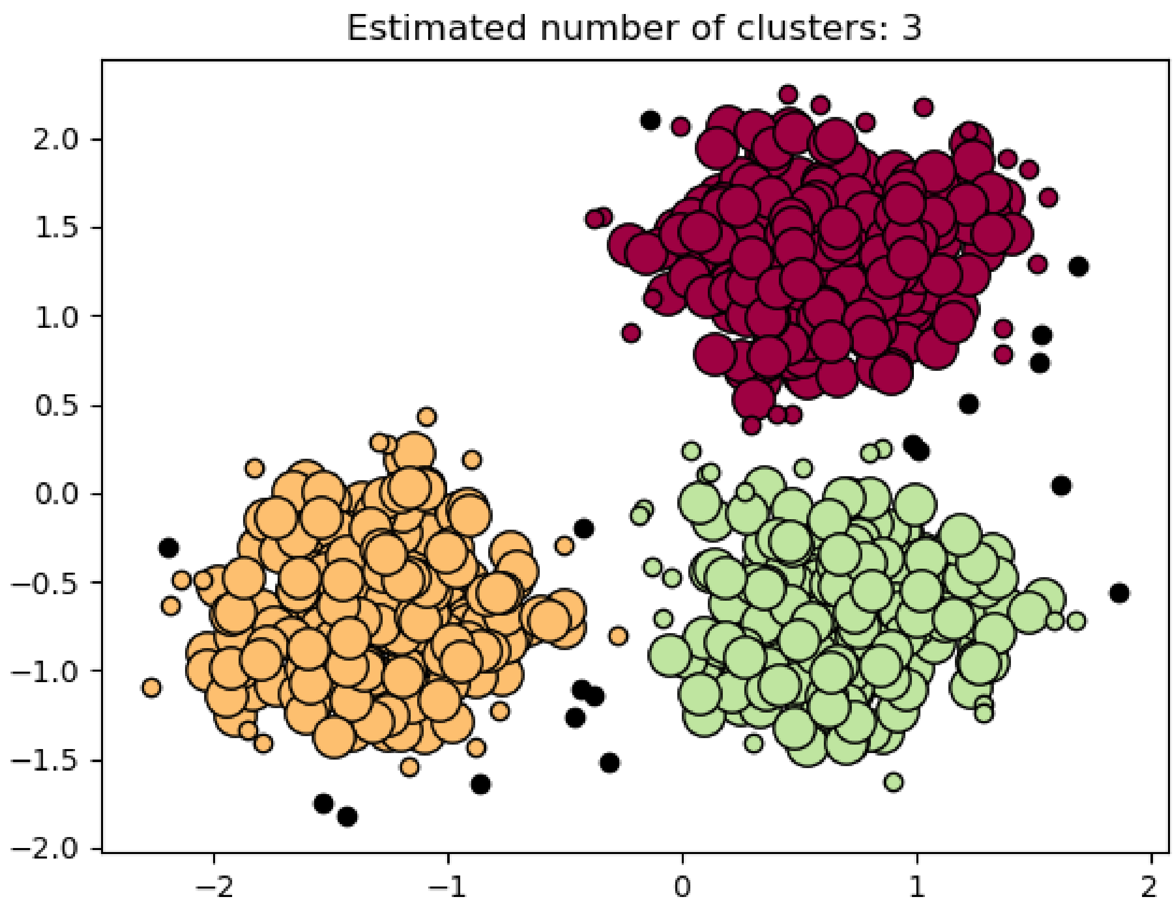
DBSCAN

- There are essentially two hyper-parameters: ϵ and P
- Points with at least P neighbors in a radius of ϵ are core points
- Points within a radius of ϵ of core points are neighbors
- Other points are outliers



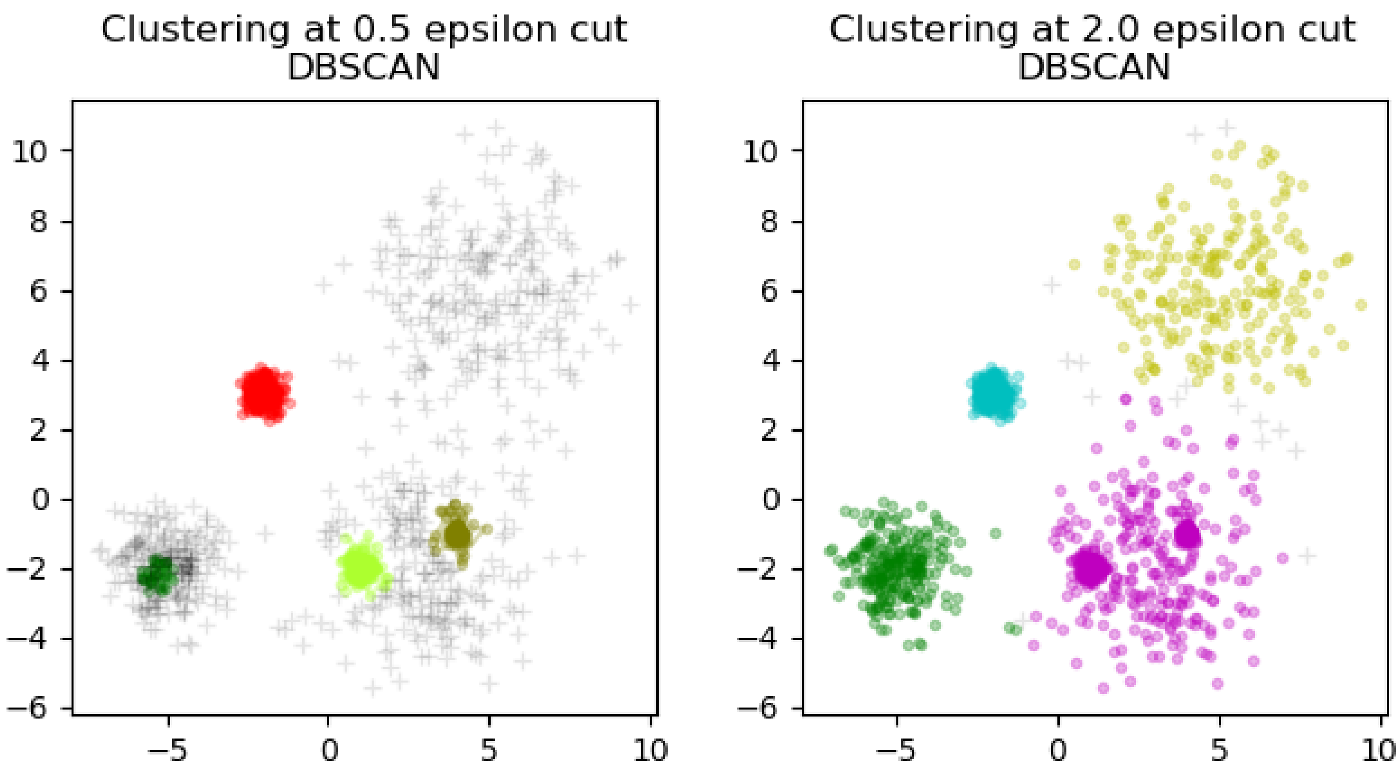
(From Wikipedia)

Illustration of DBSCAN



Limits of DBSCAN

- May not work well if the density of points varies depending on areas



Other clustering algorithms

- **Expectation-Maximization**: can be seen as a generalization of K-Means allowing for "soft" (probabilistic) assignments to non isotropic clusters
- **OPTICS**: can be seen as a generalization of DBSCAN allowing for varying point densities
- **Spectral clustering**: performs dimensionality reduction on a pairwise affinity matrix, and performs clustering in lower dimension
- …