Data Mining and Machine Learning

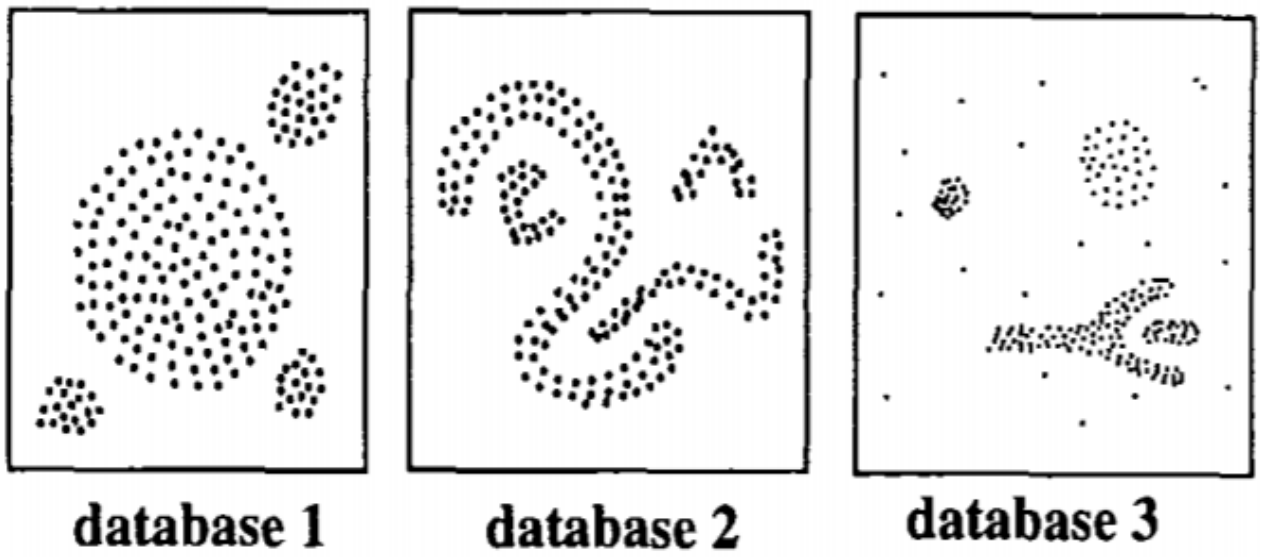
Clustering II

Dr. Edgar Acuna
Mathematical Sciences Department

Universidad de Puerto Rico- Mayaguez
academic.uprm.edu/eacuna

Density-based Approaches

- Why Density-Based Clustering methods?
  - Discover clusters of arbitrary shape.
  - Clusters – Dense regions of objects separated by regions of low density



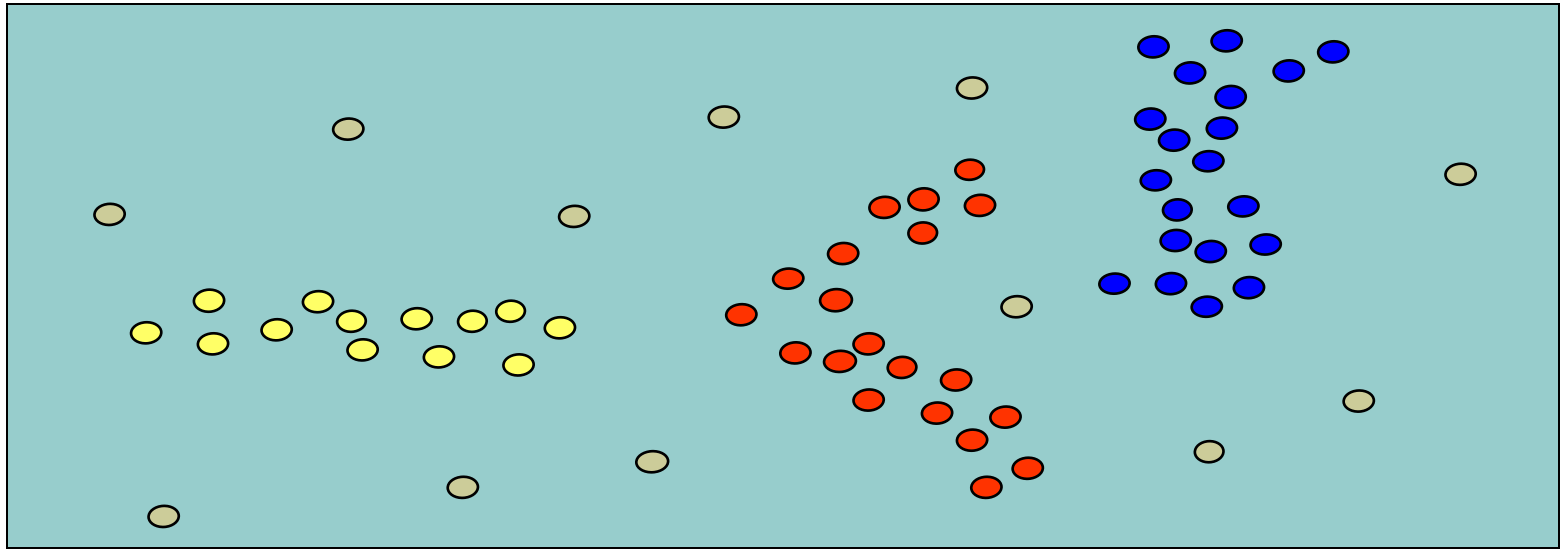
DBSCAN: Density Based Spatial Clustering of Applications with Noise

- Proposed by Ester, Kriegel, Sander, and Xu (KDD96)
- Relies on a density-based notion of cluster: A cluster is defined as a maximal set of density-connected points.
- Discovers clusters of arbitrary shape in spatial databases with noise

Density-Based Clustering

- *Basic Idea*:

  Clusters are dense regions in the data space, separated by regions of lower object density



Density Based Clustering: Basic Concept

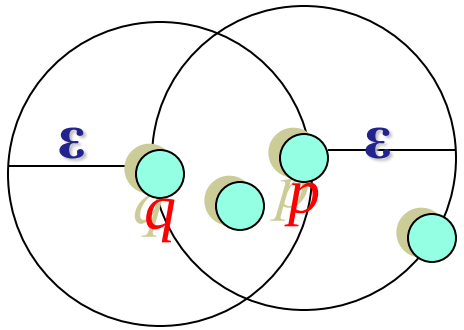
- Intuition for the formalization of the basic idea
  - For any point in a cluster, the local point density around that point has to exceed some threshold
  - The set of points from one cluster is spatially connected
- Local point density at a point *p* defined by two parameters
  - ε – radius for the neighborhood of point p: $N_{\varepsilon}(p) := \{q \text{ in data set } D \mid dist(p, q) \leq \varepsilon\}$
  - *MinPts* – minimum number of points in the given neighbourhood $N(p)$

ε-Neighborhood

- ε-Neighborhood – Objects within a radius of ε from an object.

$$N_\varepsilon(p):\{q \mid d(p,q) \le \varepsilon\}$$

- "High density" - ε-Neighborhood of an object contains at least *MinPts* of objects.



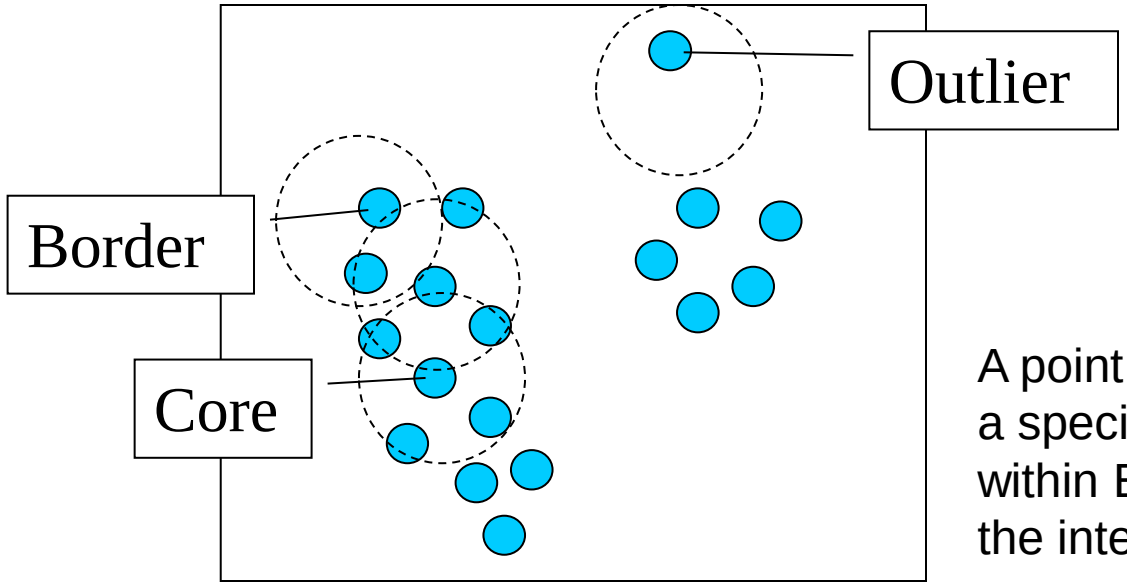
ε-Neighborhood of *p*

ε-Neighborhood of *q*

Density of p is "high" (MinPts = 4)

Density of q is "low" (MinPts = 4)

Core point, Border point & Outlier



Given ε and *MinPts*, categorize the objects into three exclusive groups.

A point is a core point if it has more than a specified number of points (MinPts) within Eps ε These are points that are at the interior of a cluster.

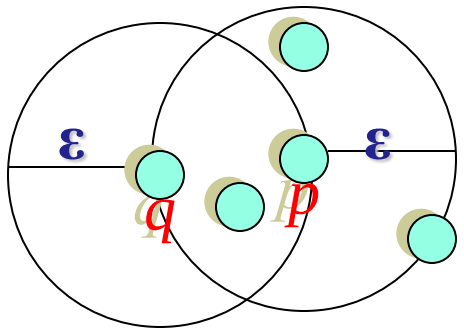
ε = 1unit, MinPts = 5

A border point has fewer than MinPts within Eps, but is in the neighborhood of a core point.

A outlier is any point that is not a core point nor a border point.

Density-Reachability

- **Directly density-reachable**
  - An object q is directly density-reachable from object p if p is a core object and q is in p's ε-neighborhood.

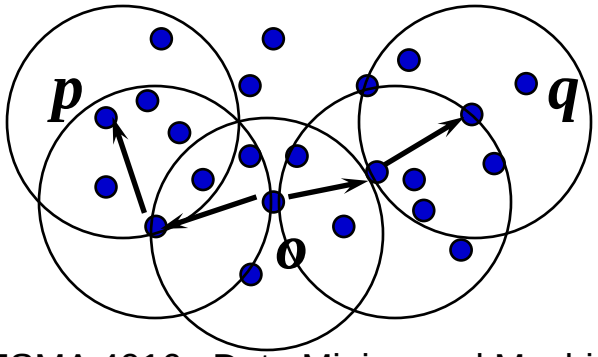


MinPts = 4

- q is directly density-reachable from p
- p is not directly density- reachable from q?
- Density-reachability is asymmetric.

Density-Connectivity

- **Density-reachable is not symmetric**
  - not good enough to describe clusters
- **Density-Connected**
  - A pair of points p and q are density-connected if they are commonly density-reachable from a point o.



- Density-connectivity is symmetric

Formal Description of Cluster

- Given a data set D, parameter ε and threshold MinPts.
- A cluster C is a subset of objects satisfying two criteria:
  - *Connected:* For all p,q ϵ C: p and q are density-connected.
  - *Maximal:* For all p,q: if p ϵ C and q is density-reachable from p, then q ϵC. (avoid redundancy)

P is a core object.

DBSCAN Algorithm

Input: The data set D

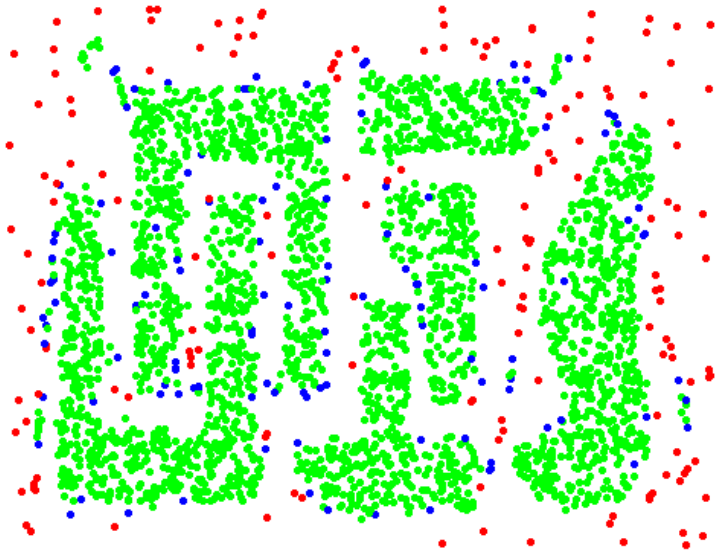
Parameter: ε, MinPts

For each object p in D
    if p is a core object and not processed then
        C = retrieve all objects density-reachable from p
        mark all objects in C as processed
        report C as a cluster
    else If *p* is a border point, no points are density-reachable from *p* and DBSCAN an continue.
    else mark p as outlier
    end if

End For

Example

Original Points

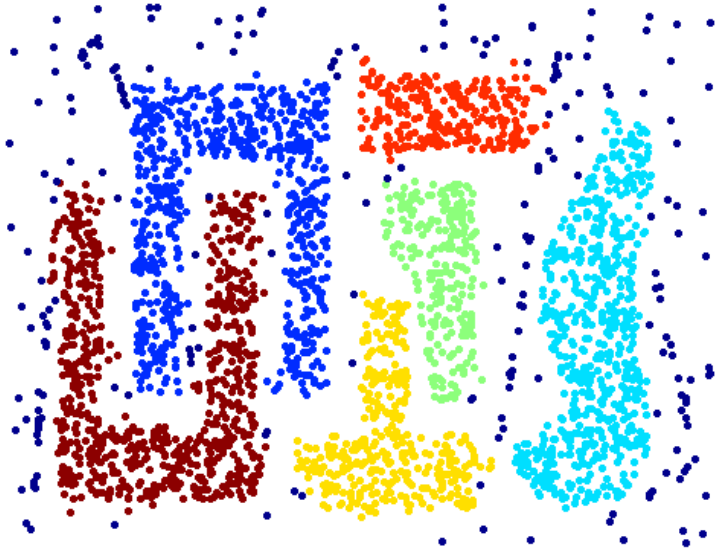


Point types: core, border and outliers

$\varepsilon = 10$, MinPts = 4

Example

Original Points

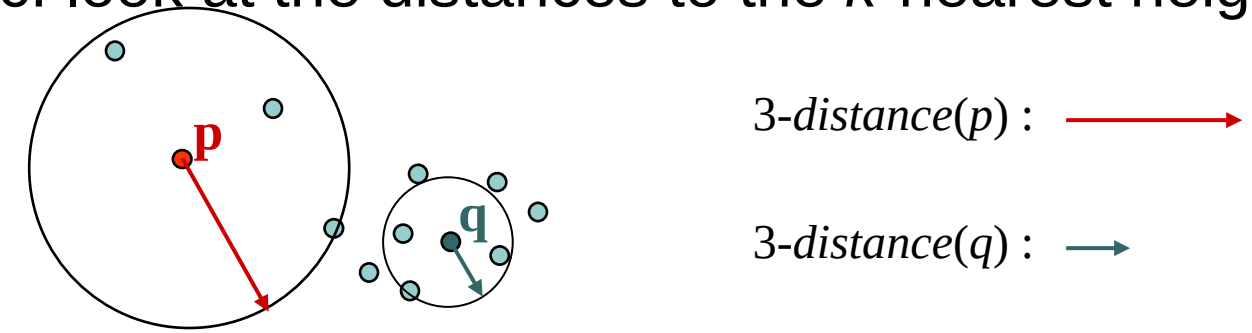


Clusters

- Resistant to Noise
- Can handle clusters of different shapes and sizes

Determining the Parameters ε and *MinPts*

- Cluster: Point density higher than specified by ε and *MinPts*
- Idea: use the point density of the least dense cluster in the data set as parameters – but how to determine this?
- Heuristic: look at the distances to the *k*-nearest neighbors

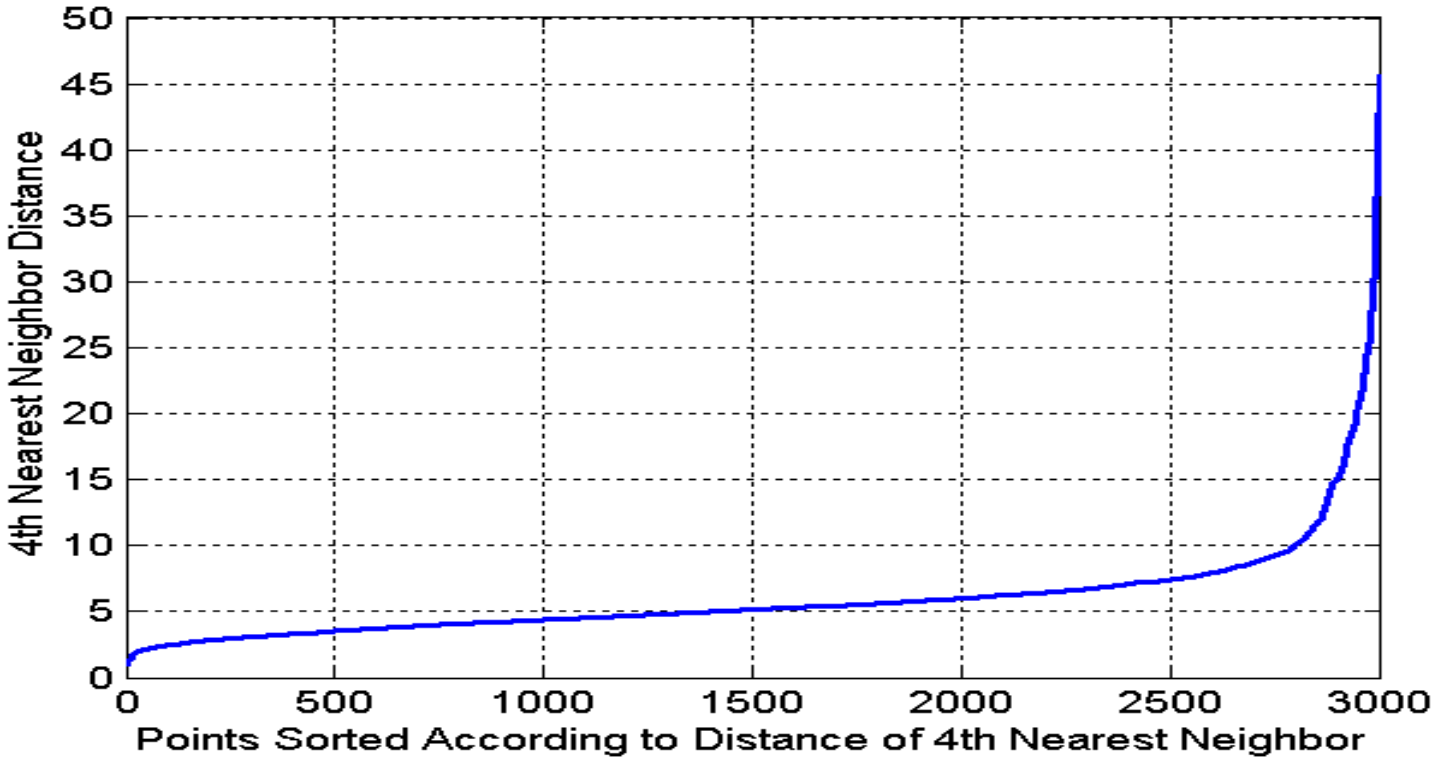


- Function *k-distance*(*p*): distance from *p* to the its *k*-nearest neighbor
- *k-distance plot*: *k*-distances of all objects, sorted in decreasing order

Determining EPS and MinPts

- Idea is that for points in a cluster, their k^{th} nearest neighbors are at roughly the same distance
- Noise points have the k^{th} nearest neighbor at farther distance
- Compute k-dist for all points for some k
- Sort them in increasing order and plot sorted values
- A sharp change at the value of k-dist that corresponds to suitable value of eps and the value of k as MinPts

Determining EPS and MinPts



MinPts=4
ε=10

Density Based Clustering: Discussion

- Advantages
  - Clusters can have arbitrary shape and size
  - Number of clusters is determined automatically
  - Can separate clusters from surrounding noise
  - Can be supported by spatial index structures
- Disadvantages
  - Input parameters may be difficult to determine
  - In some situations very sensitive to input parameter setting

Clustering based on models (Fraley and Raftery, JASA 2002)

Clustering is the grouping of objects into well-coherent groups based on characteristics, measured in. It was introduced in the late 50's by Sokal, Sneath and others, and had been developed mainly using heuristic methods. Recently it has been found that cluster analysis based on probabilistic models are useful both for understanding current methods of clustering and for suggesting new methods. The use of the model also allows us to answer questions such as, how many clusters to use ?, which cluster method is the most convenient? How to deal with the presence of outliers?

Clustering based on models

For some time, researchers have realized that cluster analysis can be carried out using probability models. With these models you are trying to see when a certain method of clustering works well.
It has been shown that some of the heuristic methods for creating conglomerates are simply approximate estimation methods of probability models. For example the k-means method and the ward method are equivalent to well-known methods that maximize approximately the classification using a normal multivariate distribution when the covariance matrix is the same for each component and proportional to the identity matrix.

Normal Multivariate Distribution

$$\phi_k(y_i / \mu_k, \Sigma_k) \equiv \frac{\exp(-\frac{1}{2}(y_i - \mu_k)^T \Sigma_k^{-1}(y_i - \mu_k))}{\sqrt{\det(2\pi\Sigma_k)}}$$

Clustering based in Finite Mixtures

Finite mixtures models have been proposed and studied often in the classification context(Wolfe, 1963, 1965,1967,1970; Edwards y Cavalli-Sforza 1965; Day 1969; Scott y Symons 1971; Duda y Hart 1973; Binder 1978).
In finite mixture models each component of the probability distribution corresponds to a cluster.
The problem of determining the number of clusters can be reformulated as a problem of model selection.
Outliers are treated by adding one component that represents a distribution for anomalous data.

Clustering based in Finite Mixtures

The likelihood function of a mixture model with G components given the random sample y_1, y_2,… y_n of the random variable and is defined by

$$L(\theta_1,....\theta_G;\tau_1,....\tau_G / y) = \prod_{i=1}^{n}\sum_{k=1}^{G}\tau_k f_k(y_i / \theta_k)$$

where f_k y θ_k are the density functions and the parameters of the k-th component of the sample and τ_k is the probability of an observation corresponding to the k-th component. τ_k is non negative and its sum must be 1.
Usually f_k is a multivariate normal density ϕ_k, paramete by its mean and covariance matrix.

Clustering based on Gaussian Mixtures

The geometric characteristics (shape, volume, orientation) of the clusters are determined by the covariances $\Sigma_{k,}$ which, can be parameterized to impose restrictions between clusters. Thus, if $\Sigma_k=\lambda I$, is considered, the all the clusters are spherical and of the same size, if $\Sigma_k=\Sigma$ then all the clusters have the same geometry but not necessarily all of them are spherical.

For the first case we need only one parameter and for the second d(d+1)/2 parameters

Using Gaussian Mixtures in Python

The following models are compared:
'full' (each component has its own general covariance matrix),
'tied' (all components share the same general covariance matrix),
'diag' (each component has its own diagonal covariance matrix),
'spherical' (each component has its own single variance).

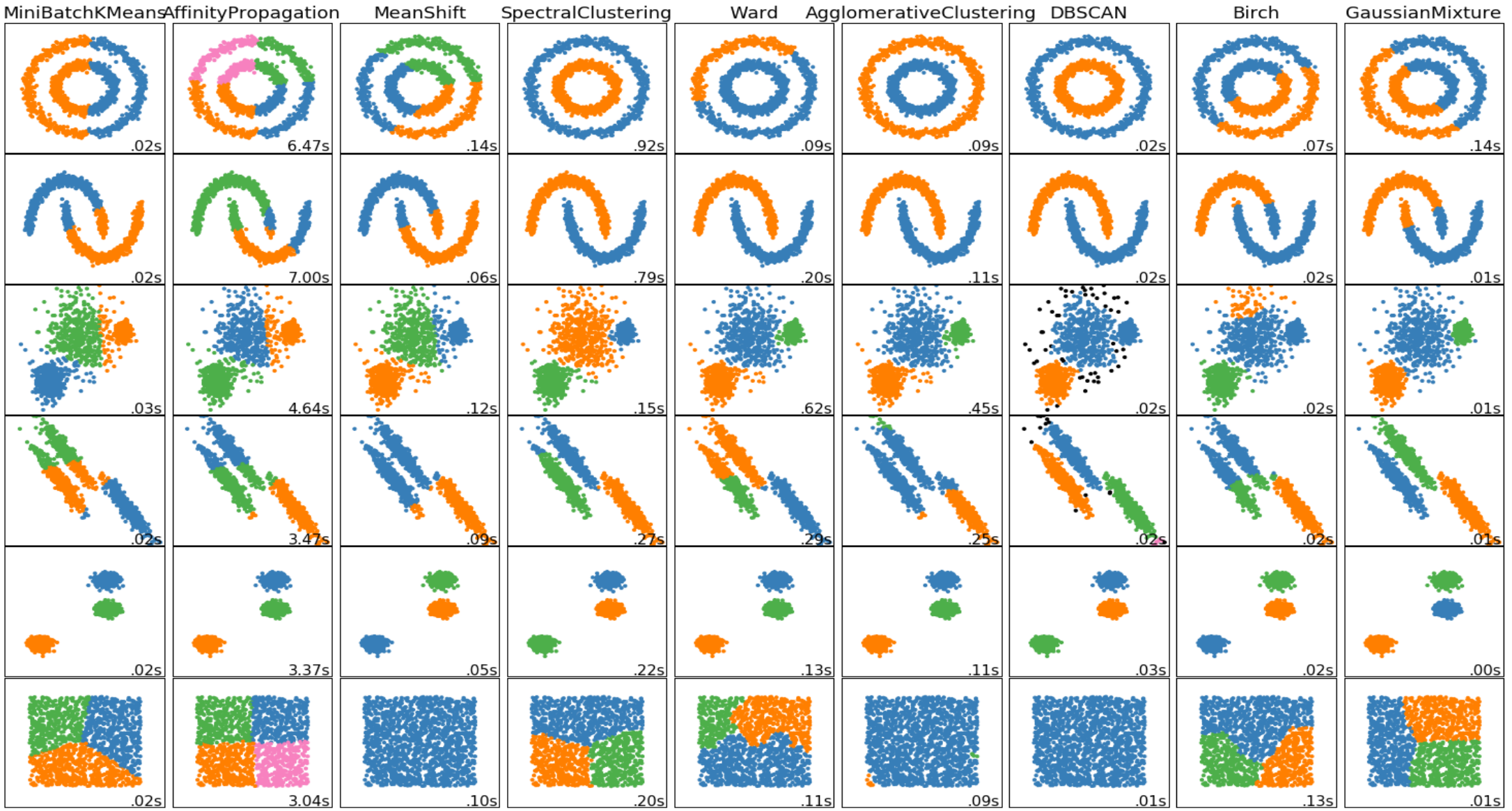
The BIC (Bayesian Information Criterion) for best model selection

$$2\log p(D/M_k) \approx 2\log p(D/\hat{\theta}_k, M_k) - v_k \log(n) = BIC_k$$

where ν_k is the number of indepent parameters to be estimated with the M_K model (Schwarz 1978).
The best model will be the one with the biggest BIC.

Other clustering methods in Python

Mini Batch Kmeans
Affinity Propagation (it requires a lot of computing time)
Mean-Shift
Spectral Clustering
Birch


MiniBatchKMeans
AffinityPropagation
MeanShift
SpectralClustering
Ward
AgglomerativeClustering
DBSCAN
Birch
GaussianMixture
.02s
6.47s
.14s
.92s
.09s
.09s
.02s
.07s
.14s
.02s
7.00s
.06s
.79s
.20s
.11s
.02s
.02s
.01s
.03s
4.64s
.12s
.15s
.62s
.45s
.02s
.02s
.01s
.02s
3.47s
.09s
.27s
.29s
.25s
.02s
.02s
.01s
.02s
3.37s
.05s
.22s
.13s
.11s
.03s
.02s
.00s
.02s
3.04s
.10s
.20s
.11s
.09s
.01s
.13s
.01s