Density-based clustering

Used in many applications

- Mainly for its efficiency, resistance to noise and ability to deal with arbitrary shaped clusters

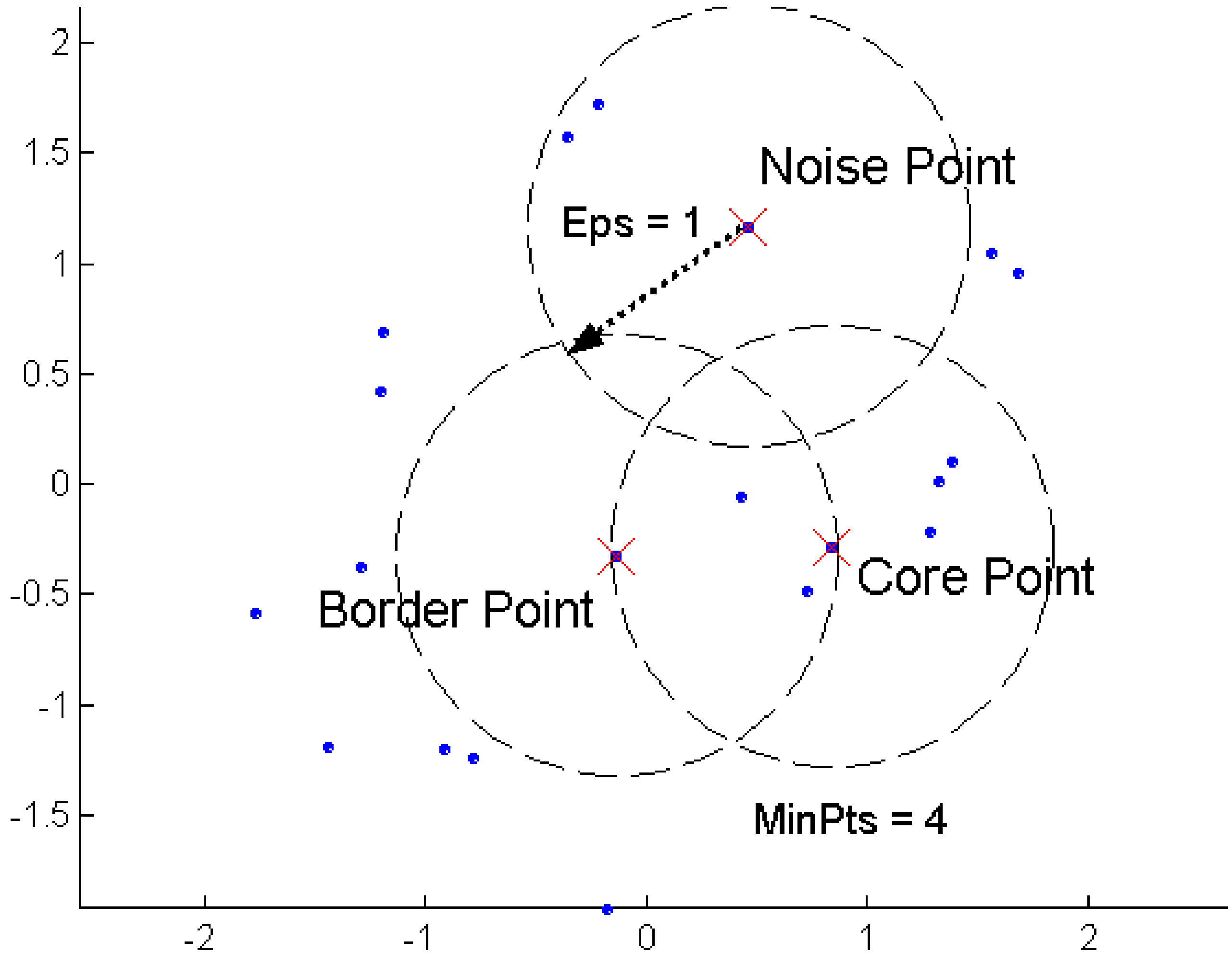
Main idea: divide noise from objects to clusters

- Objects to cluster = dense points
- Noise = low-density points

DBSCAN

- DBSCAN is a density-based algorithm.
  - Density = number of points within a specified radius (Eps)
  - A point is a core point if it has more than a specified number of points (MinPts) within Eps
    - These are points that are at the interior of a cluster
  - A border point has fewer than MinPts within Eps, but is in the neighborhood of a core point
  - A noise point is any point that is not a core point or a border point.

DBSCAN: Core, Border, and Noise Points



DBSCAN Algorithm

- Eliminate noise points
- Perform clustering on the remaining points

Algorithm 8.4 DBSCAN algorithm.

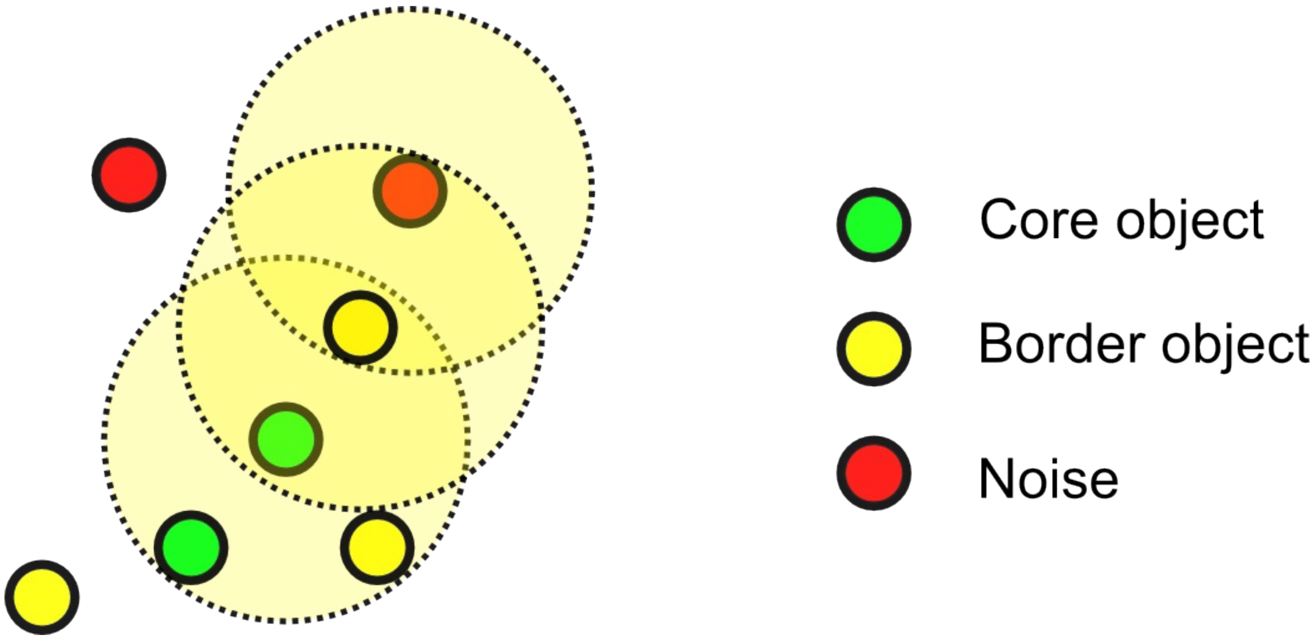
1: Label all points as core, border, or noise points.
2: Eliminate noise points.
3: Put an edge between all core points that are within Eps of each other.
4: Make each group of connected core points into a separate cluster.
5: Assign each border point to one of the clusters of its associated core points.

Border points can be neighbors of several core points/clusters
→ arbitrarily choose one!

DBSCAN

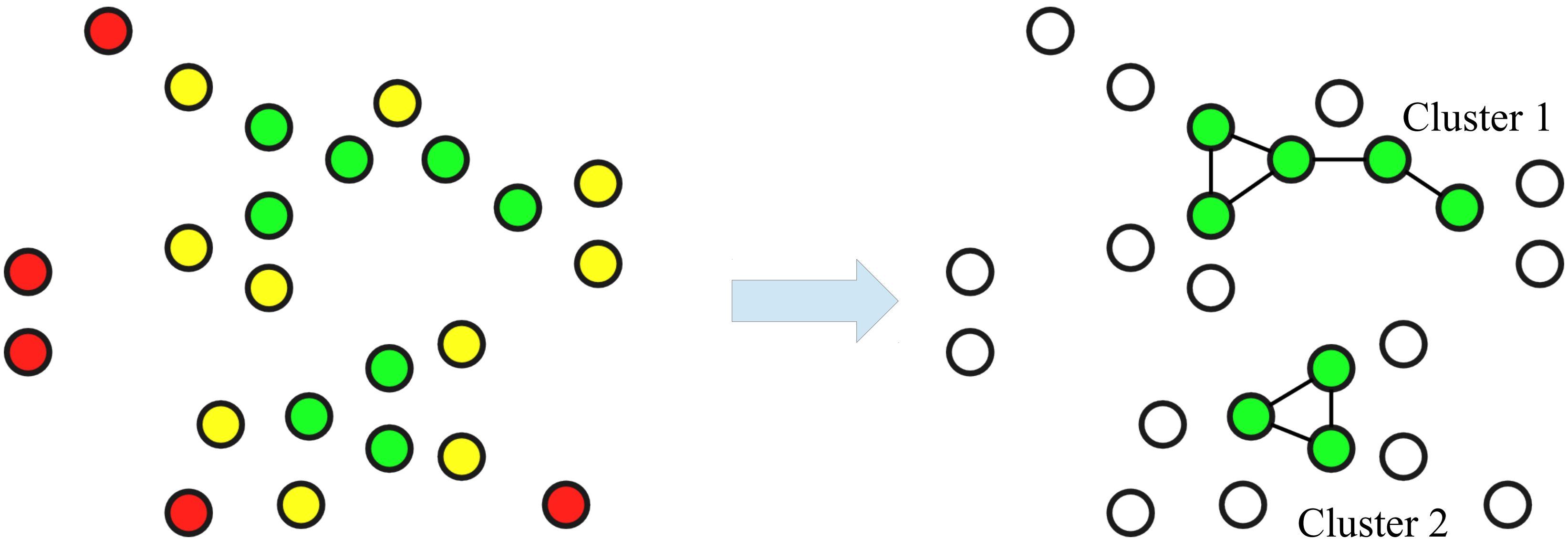
Step 1: label points as core (dense), border and noise

- Based on thresholds R (radius of neighborhood) and min_pts (min number of neighbors)



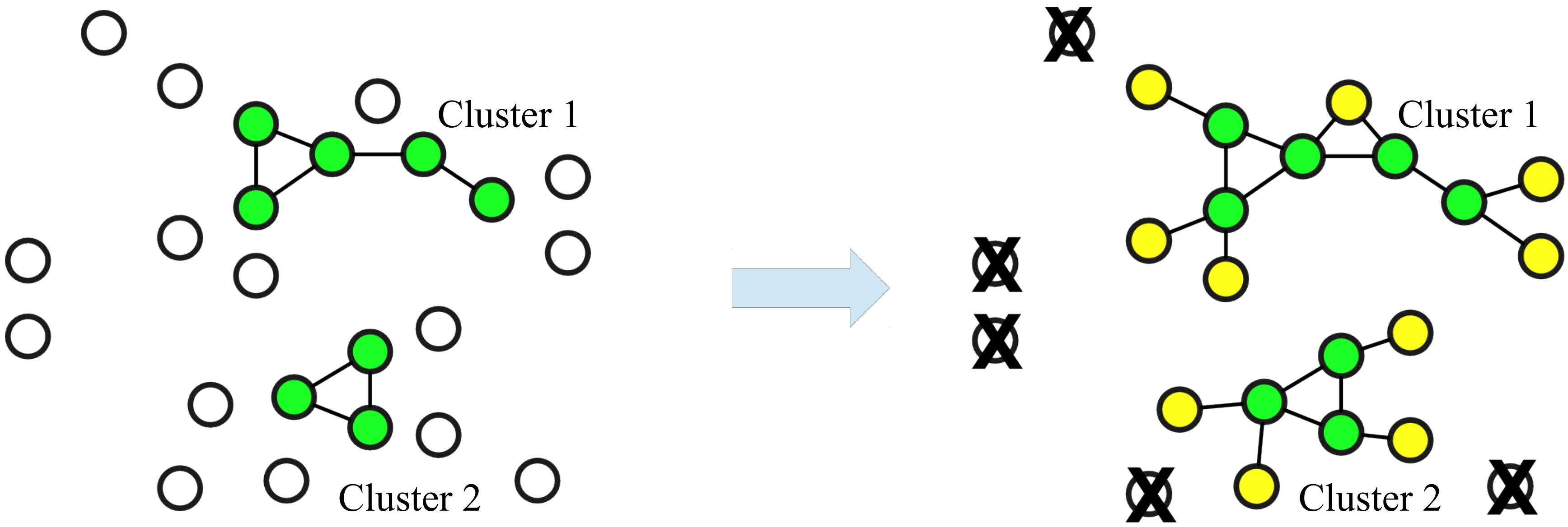
DBSCAN

Step 2: connect core objects that are neighbors, and put them in the same cluster



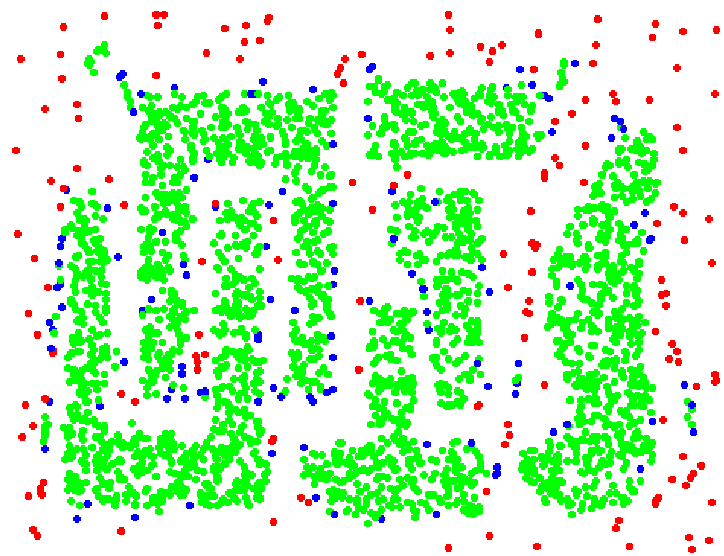
DBSCAN

Step 3: associate border objects to (one of) their core(s), and remove noise



DBSCAN: Core, Border and Noise Points

Original Points

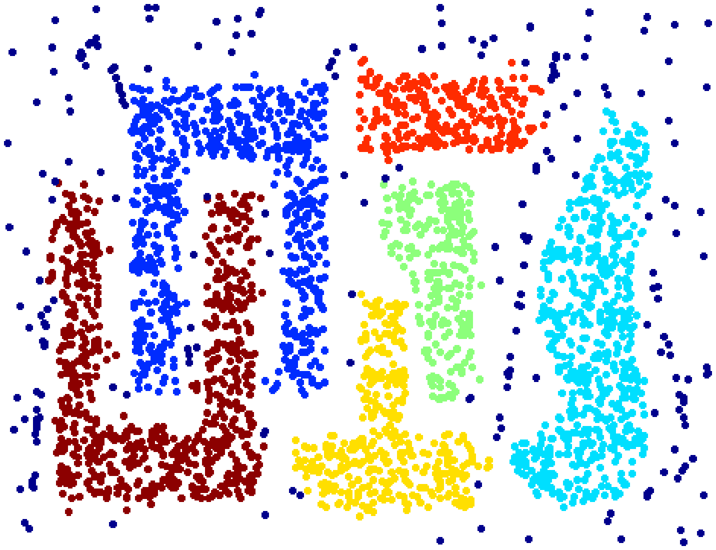


Point types: core, border and noise

Eps = 10, MinPts = 4

When DBSCAN Works Well

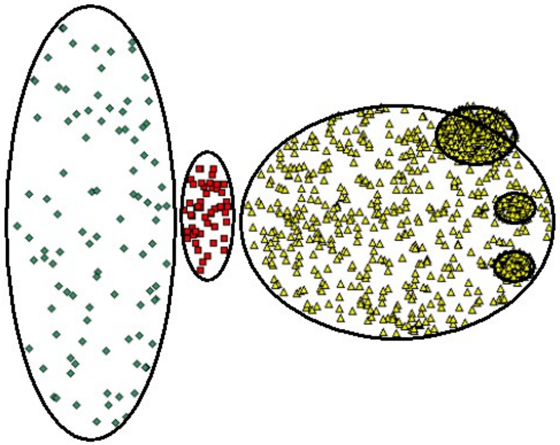
Original Points



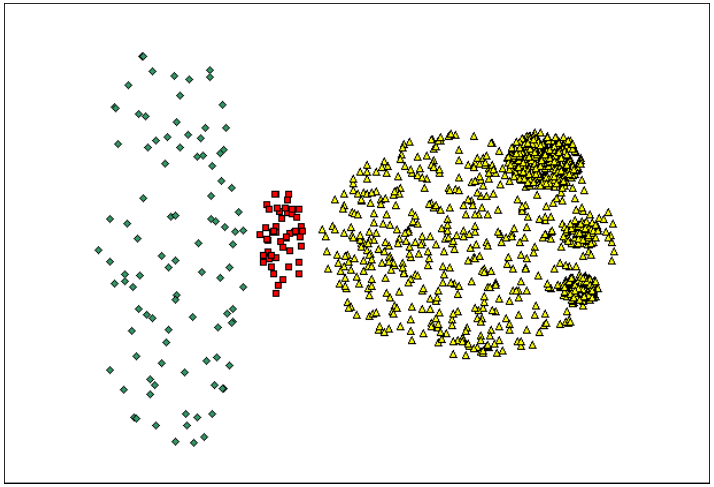
Clusters

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

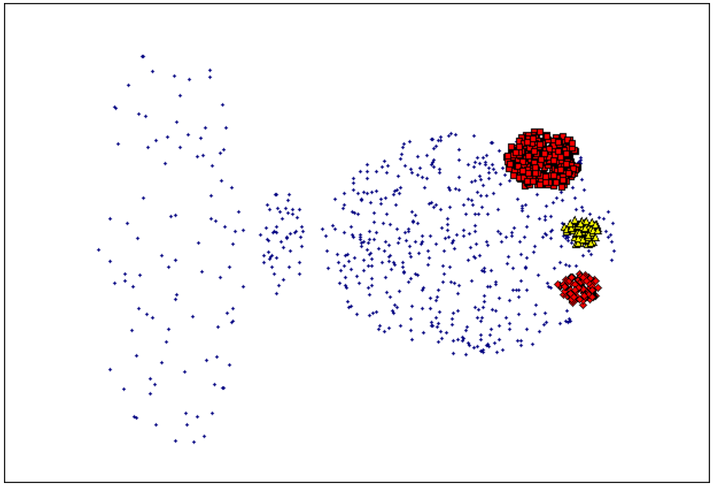
When DBSCAN Does NOT Work Well



Original Points



(MinPts=4, Eps=9.92)

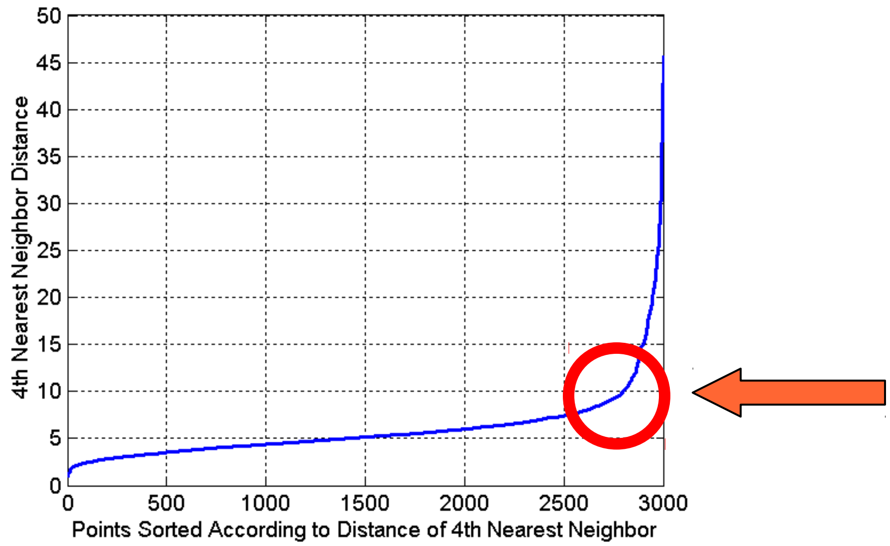


(MinPts=4, Eps=9.75).

- **Varying densities**
- **High-dimensional data**

DBSCAN: 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
- So, plot sorted distance of every point to its k[th] nearest neighbor

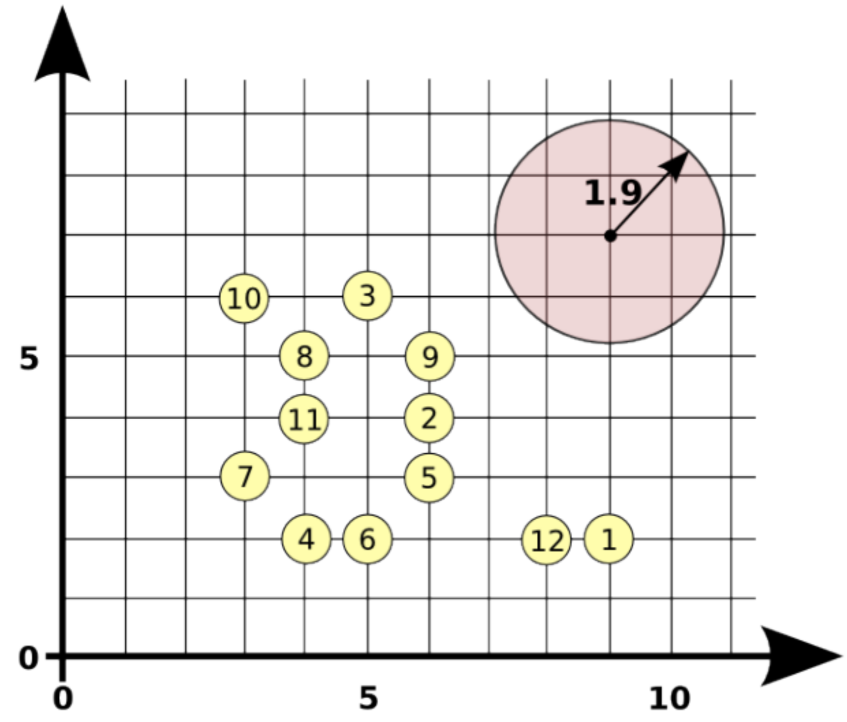


Exercises

Sul seguente dataset:

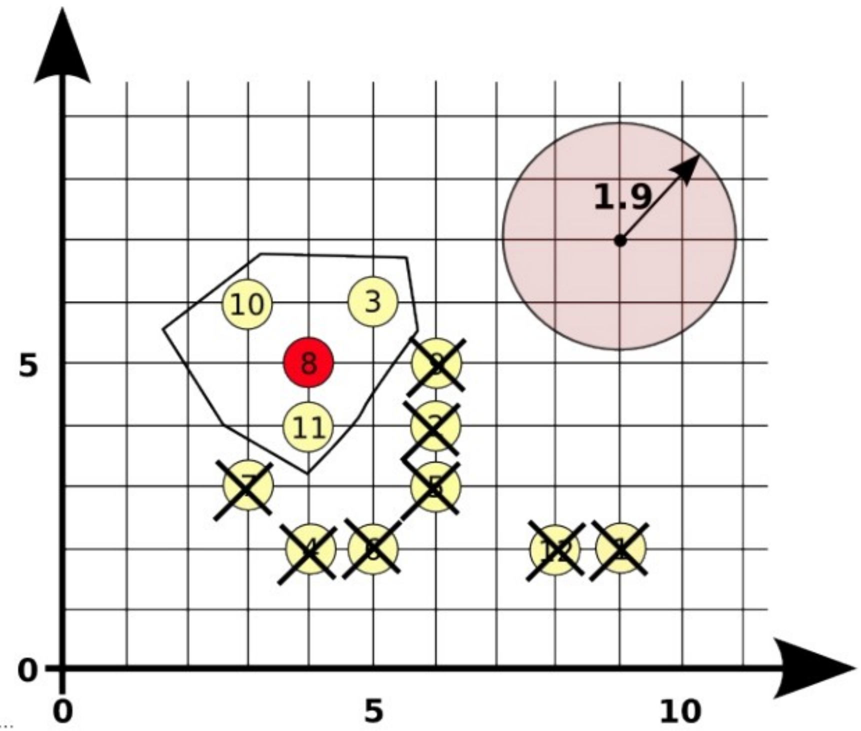
A) Si utilizzi l'algoritmo di clustering density-based DBSCAN, con raggio (ε) pari a 1.9, e minPts pari a 4 (=3 vicini + il punto di cui si calcola la densità). Si richiede di (1) indicare il numero di cluster che si ottengono; (2) per ogni punto indicare il cluster di appartenenza; (3) per ogni punto dire se si tratta di un *core point*, *border point* o *rumore*. **(8 punti)**

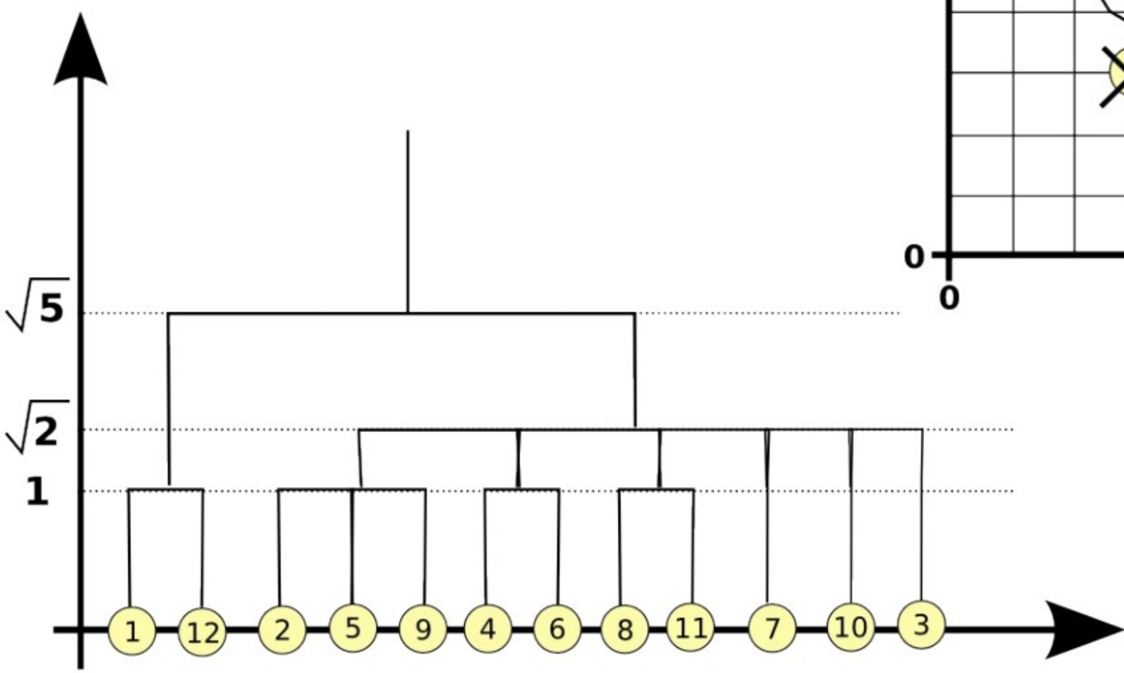
B) Si disegni il dendogramma ottenuto con un algoritmo di clustering agglomerativo MIN-link (o *Single linkage*). **(4 punti)**



Exercises

Solution:



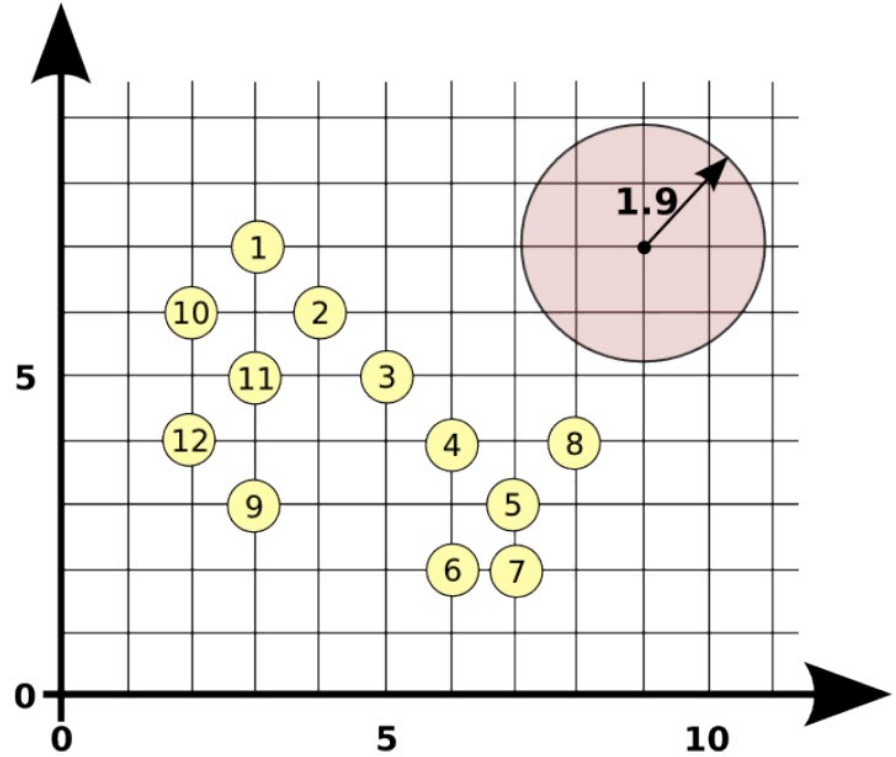


Exercises

Sul seguente dataset:

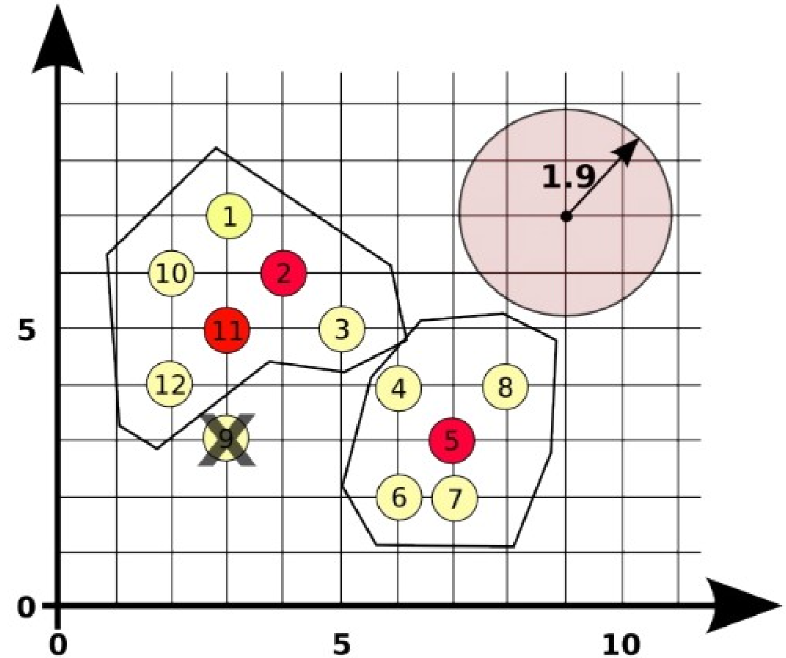
A) Si utilizzi l'algoritmo di clustering density-based DBSCAN, con raggio (ε) pari a 1.9, e minPts pari a 4 (=3 vicini + il punto di cui si calcola la densità).
(1) per ogni punto dire se si tratta di un *core point*, *border point* o *rumore;*
(2) indicare la composizione dei cluster ottenuti. **(5 punti)**

B) Simulare l'esecuzione dell'algoritmo k-means sullo stesso insieme di punti, con k=2 e centri iniziali $c_1=(3,7)$ e $c_2=(8,2)$. **(5 punti)**

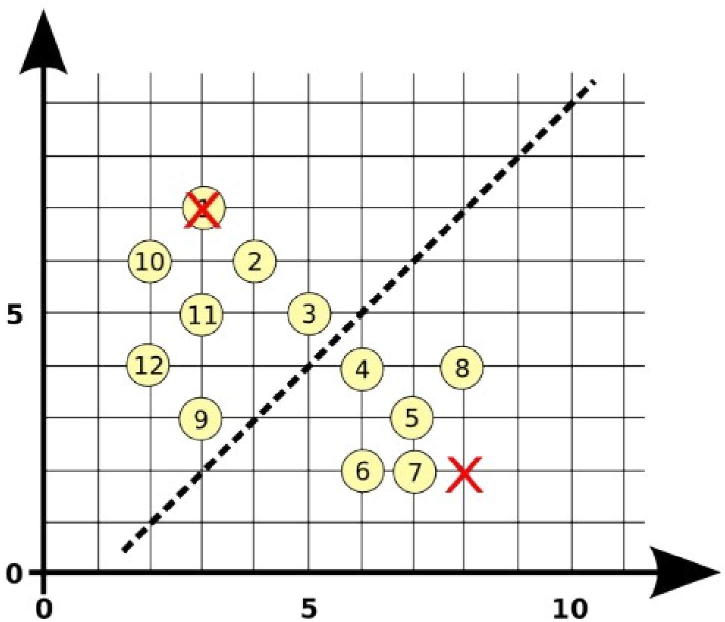


Exercises

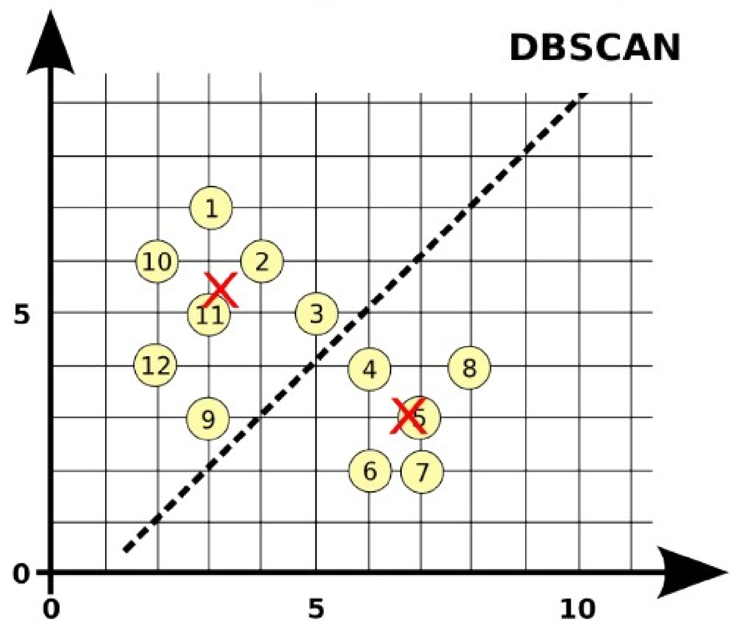
Solution:



DBSCAN



K-means - iterazione 1



K-means - iterazione 2
(convergenza)

Exercises

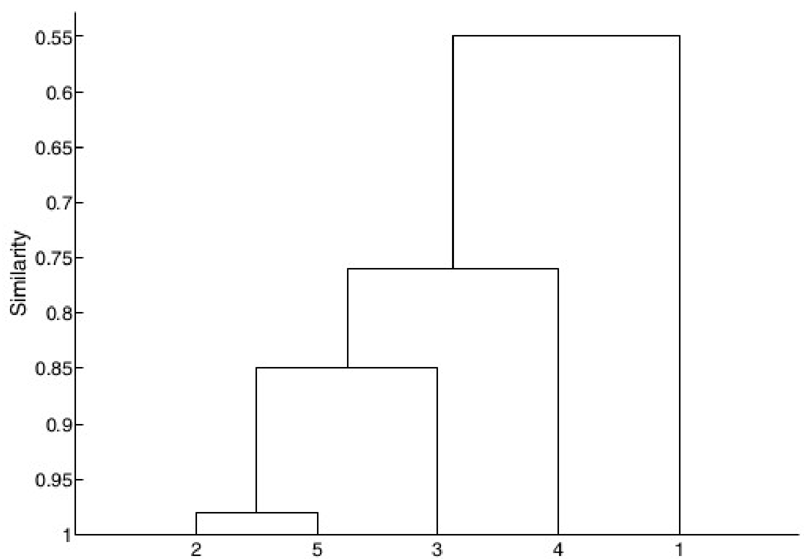
Execute single-linkage and complete-linkage HAC on the following **similarity** matrix, and draw the corresponding dendograms:

	p1	p2	p3	p4	p5
p1	1.00	0.10	0.41	0.55	0.35
p2	0.10	1.00	0.64	0.47	0.98
p3	0.41	0.64	1.00	0.44	0.85
p4	0.55	0.47	0.44	1.00	0.76
p5	0.35	0.98	0.85	0.76	1.00

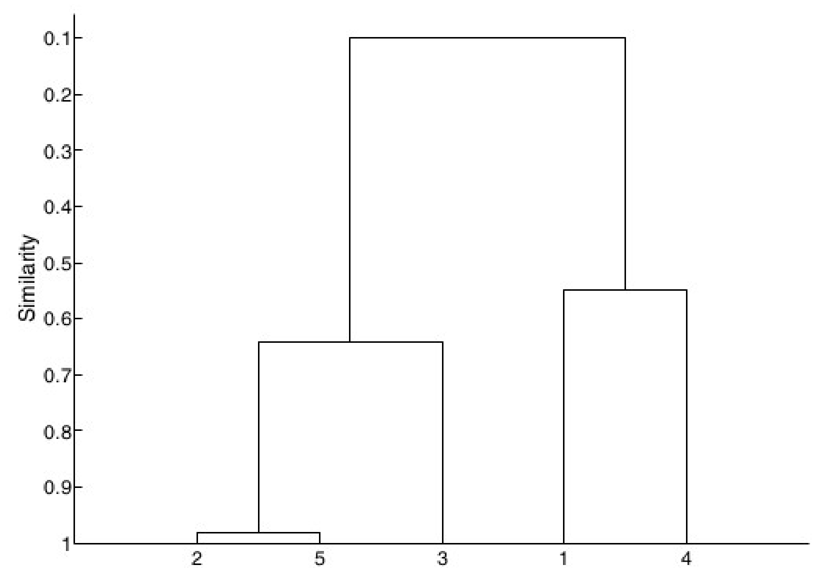
Exercises

Solution:

	p1	p2	p3	p4	p5
p1	1.00	0.10	0.41	0.55	0.35
p2	0.10	1.00	0.64	0.47	0.98
p3	0.41	0.64	1.00	0.44	0.85
p4	0.55	0.47	0.44	1.00	0.76
p5	0.35	0.98	0.85	0.76	1.00



(a) Single link.



(b) Complete link.