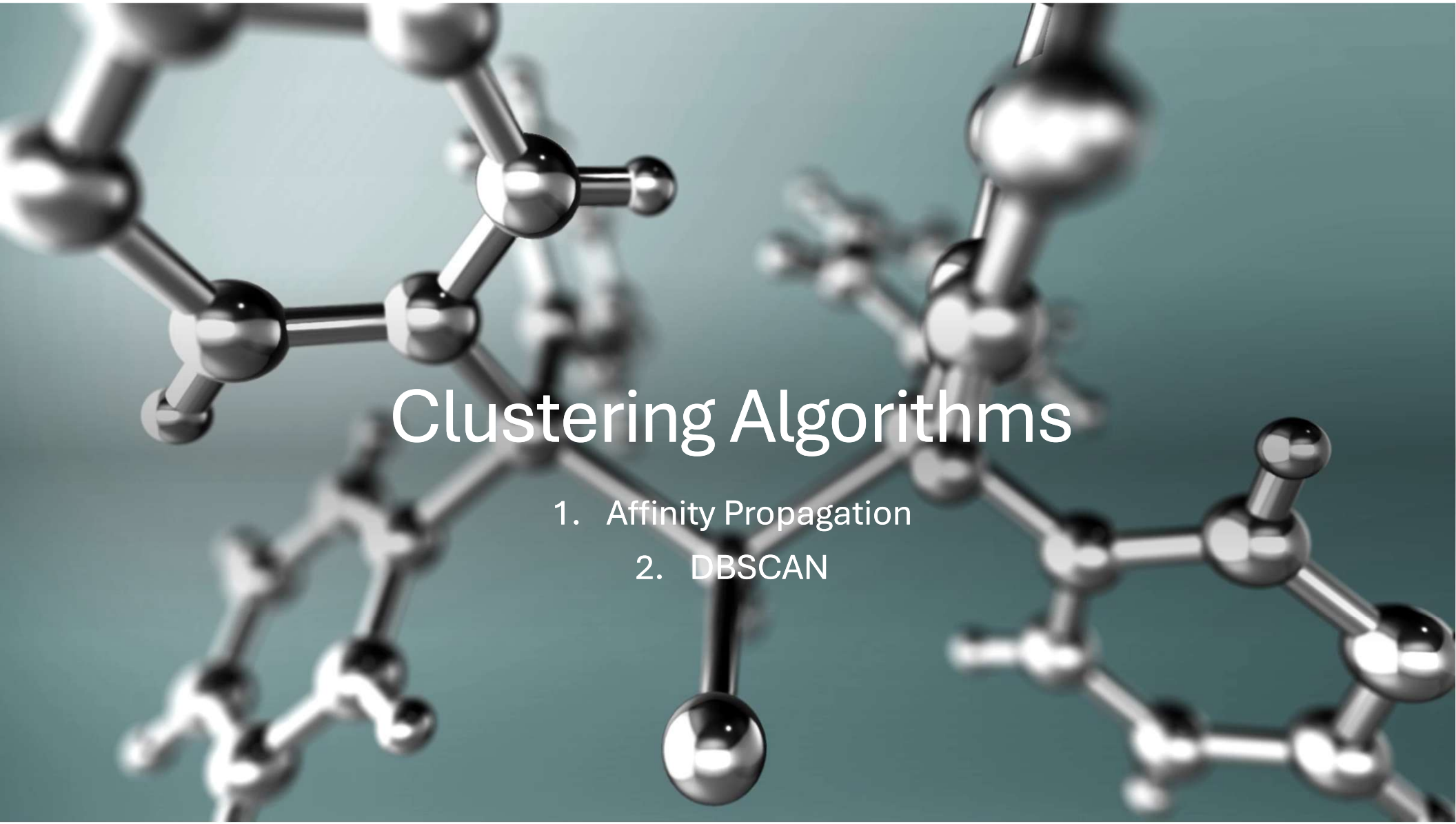


Clustering Algorithms

1. Affinity Propagation
2. DBSCAN

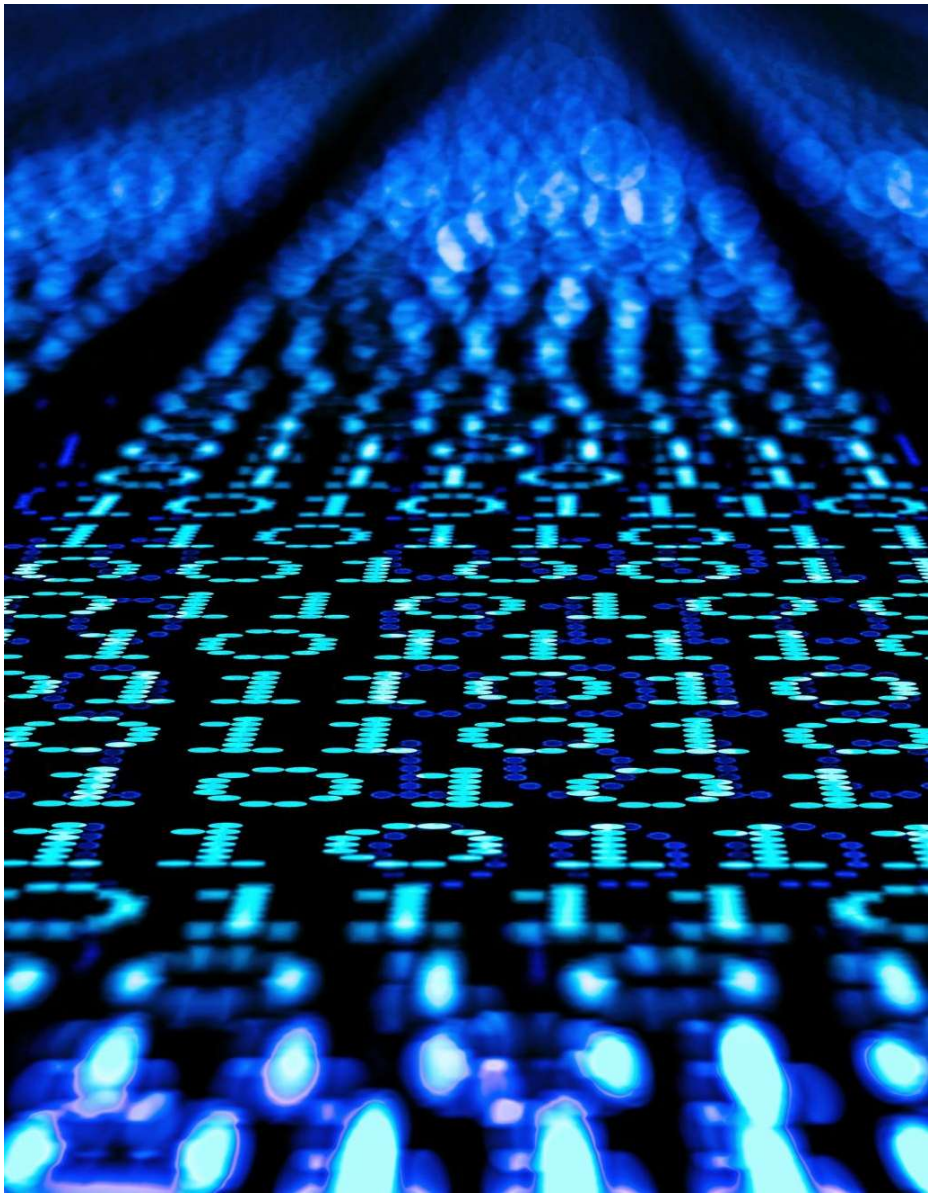
AFFINITY PROPAGATION

Affinity – Rapport, Harmony

Propagation- transmission or distribution

Affinity Propagation

- An exemplar-based clustering method
- Identifies the exemplars within the dataset by sending messages between datasets
- An exemplar is a representative data point that best represents a cluster
- It doesn’t require specifying the number of clusters beforehand. Instead, it identifies optimal number of clusters automatically based on the data.



- *Similarity matrix:*

Similarity matrix is computed by using the below formula.

$$S(i,j)=-\|xi-xj\|2$$

Diagonal elements S(i,i) are called preferences. A higher preference is more likely to become a cluster center.

- *Responsibility Update:*

The Responsibility matrix R reflects how suitable a point x_k *is to be the exemplar for point xi*, relative to other candidates.

$$r(i,k)\leftarrow s(i,k)-\max k'\neq k\{a(i,k')+s(i,k')\}$$

A high $r(i, k)$ means x_k is much more suitable than any other candidate to represent x_i.

- Availability update:

The availability matrix A represents how appropriate it is for
x_i to choose x_k as its exemplar, considering the preferences of other points:

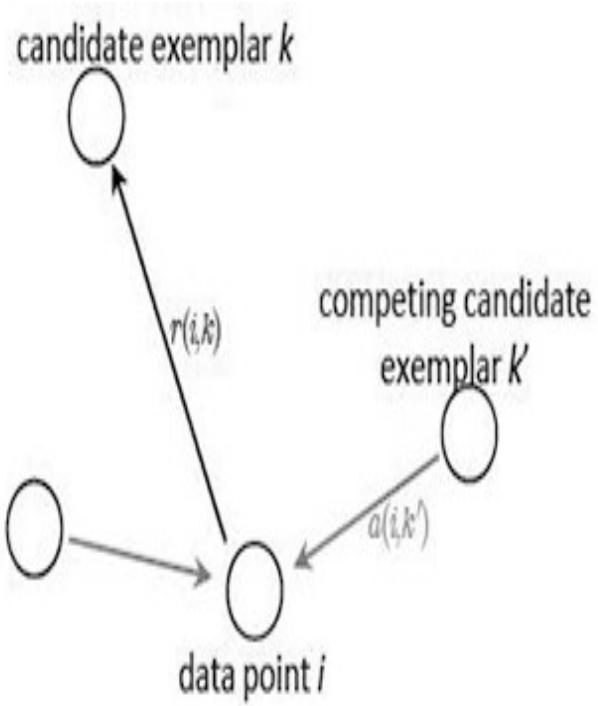
For$i \neq k$:

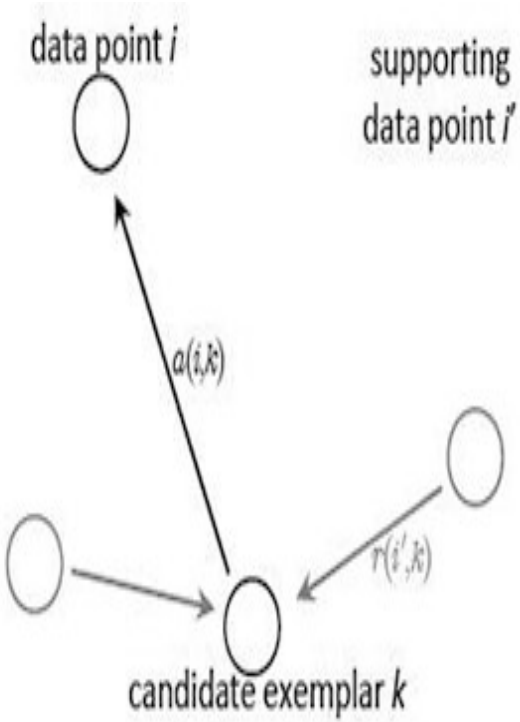
$$a(i, k) \leftarrow \min(0, r(k, k) + \sum_{i' \neq \{i,k\}} \max(0, r(i', k)))$$

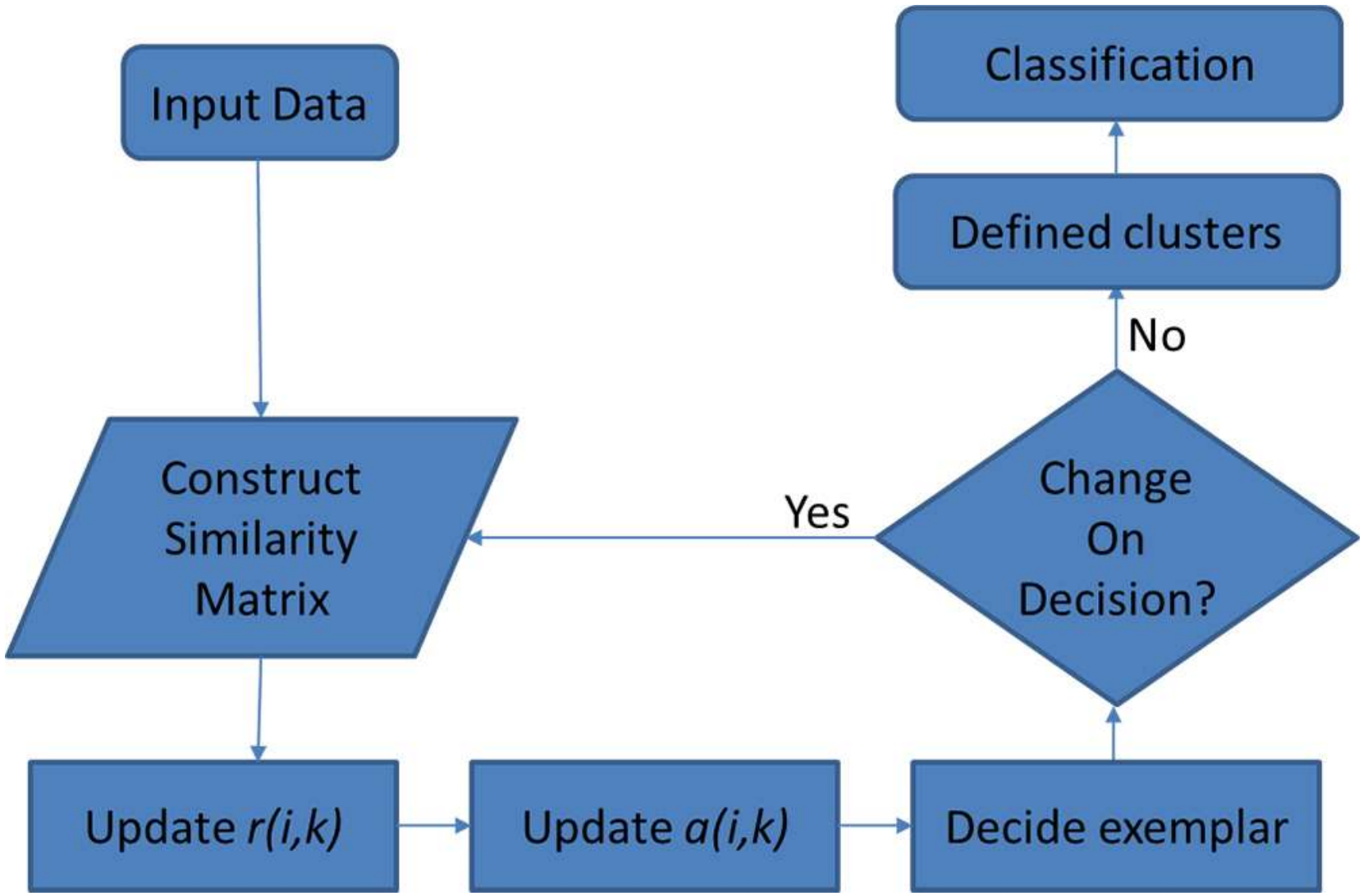
For $i = k$ (self-availability):

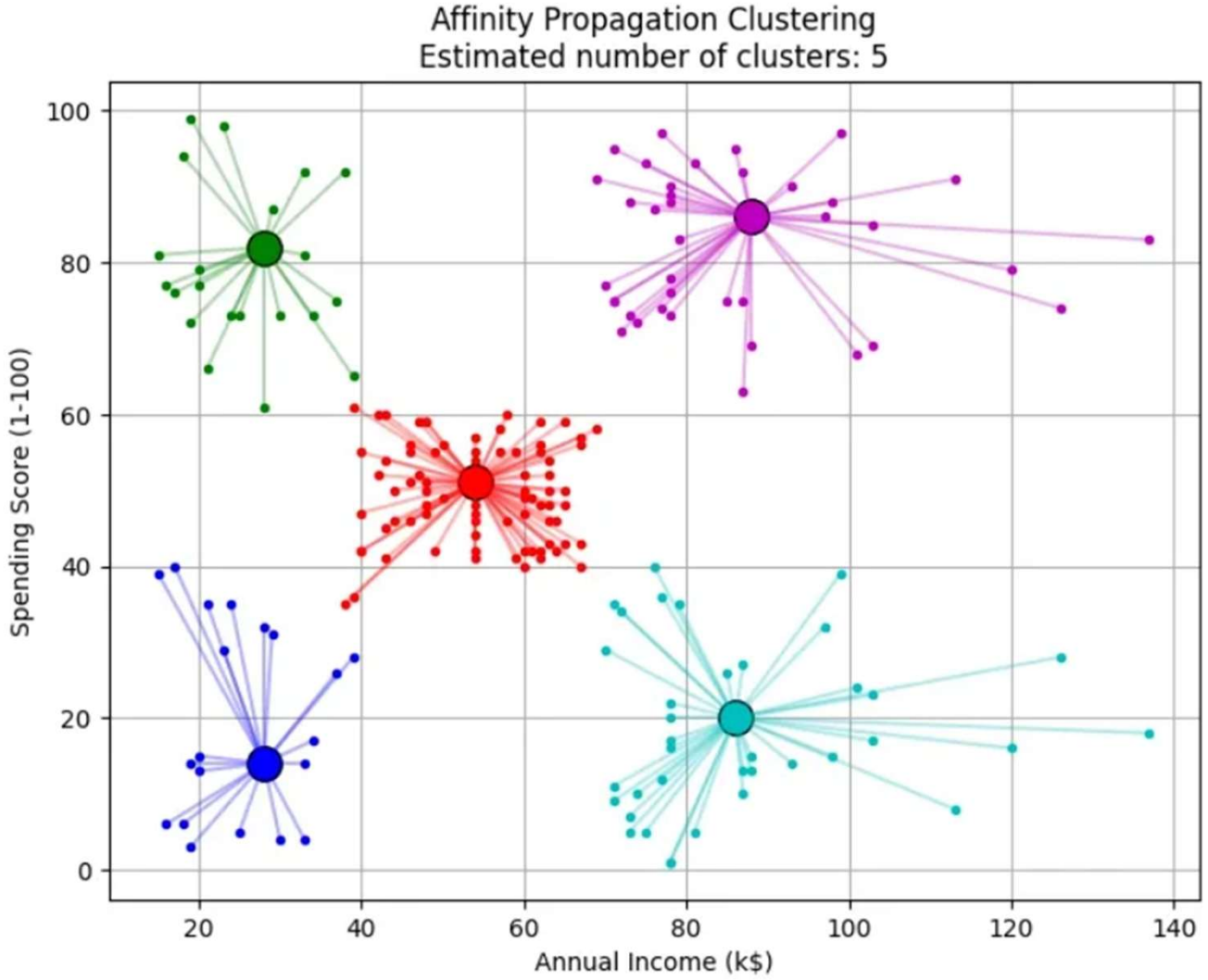
$$a(k, k) \leftarrow \sum_{i' \neq k} \max(0, r(i', k))$$

Responsibility reflects a candidate's suitability; availability reflects the support from other points.


candidate exemplar k
r(i,k)
competing candidate
exemplar k'
a(i,k')
data point i

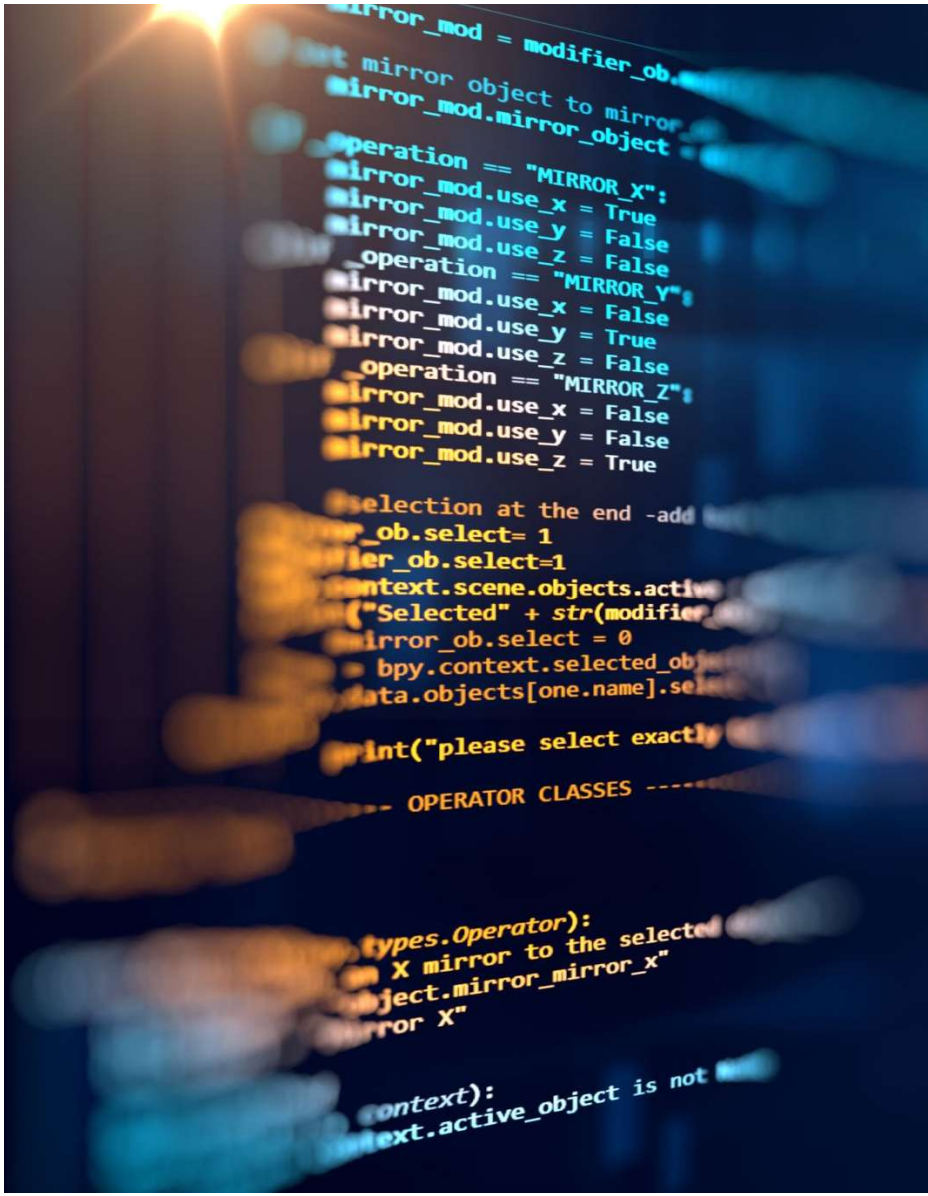

data point i
supporting
data point i'
a(i,k)
r(i',k)
candidate exemplar k

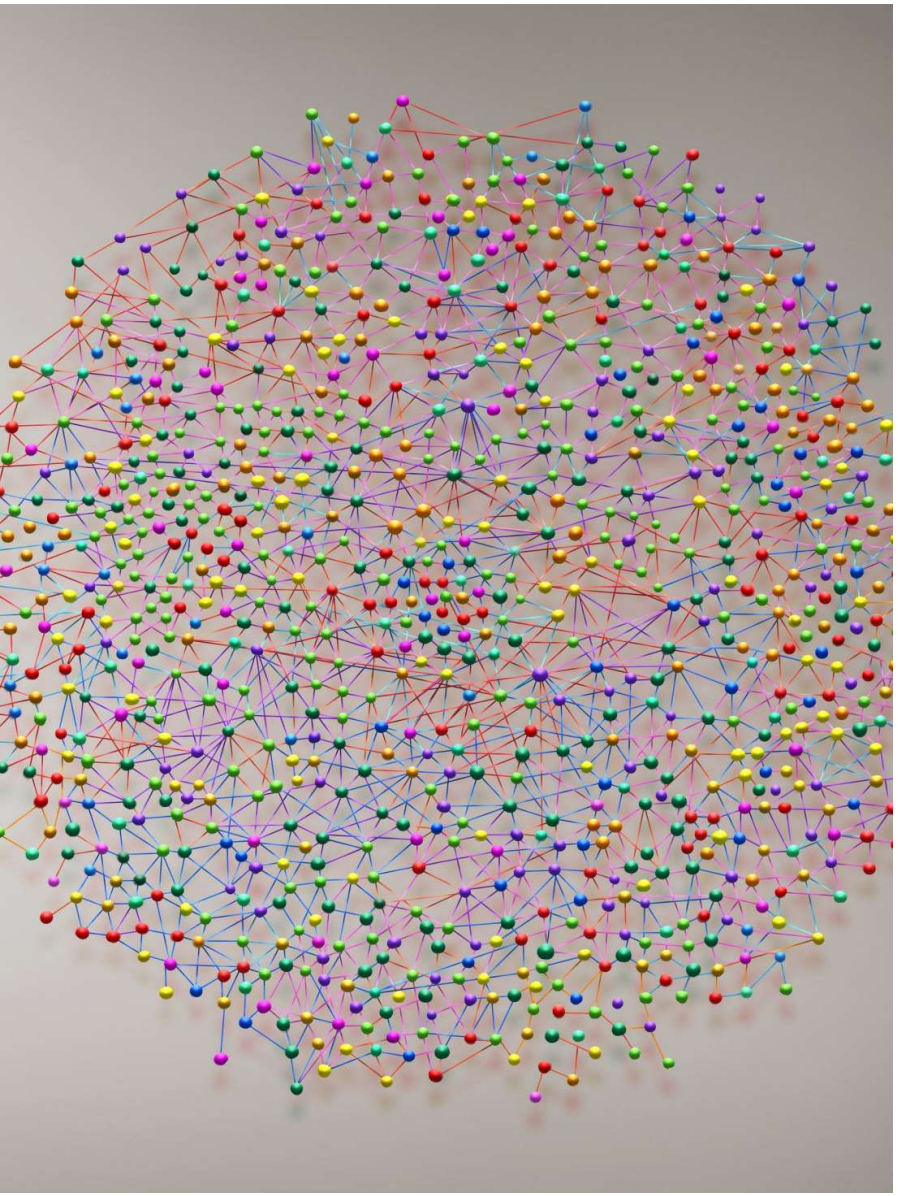

Input Data
Classification
Defined clusters
No
Construct Similarity Matrix
Yes
Change On Decision?
Update r(i,k)
Update a(i,k)
Decide exemplar


Affinity Propagation Clustering
Estimated number of clusters: 5
Spending Score (1-100)
Annual Income (k$)
0
20
40
60
80
100
20
40
60
80
100
120
140

Parameters

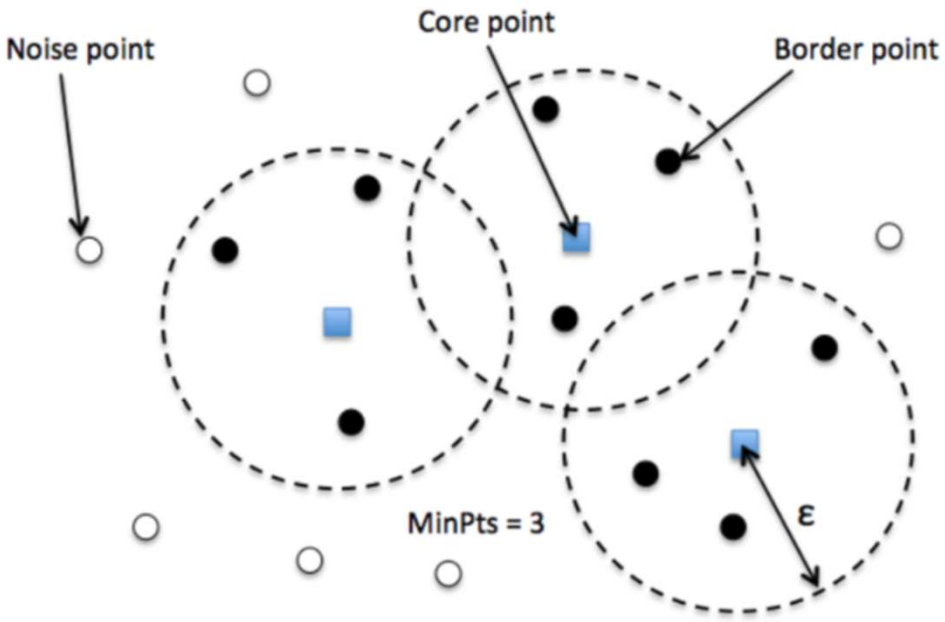
- Preference – It controls the number exemplars(cluster centers) chosen by the algorithm
- Damping- The damping factor helps stabilize the algorithm by limiting how much each update can change between iterations.
- Max_iter: This parameter determines the maximum number of iterations and passes over data.
- Convergence_iter: This parameter controls the number of iterations with no change in the cluster assignments which must occur for the algorithm to consider the solution as converged.
- Random_state: This parameter handles the randomness of the data



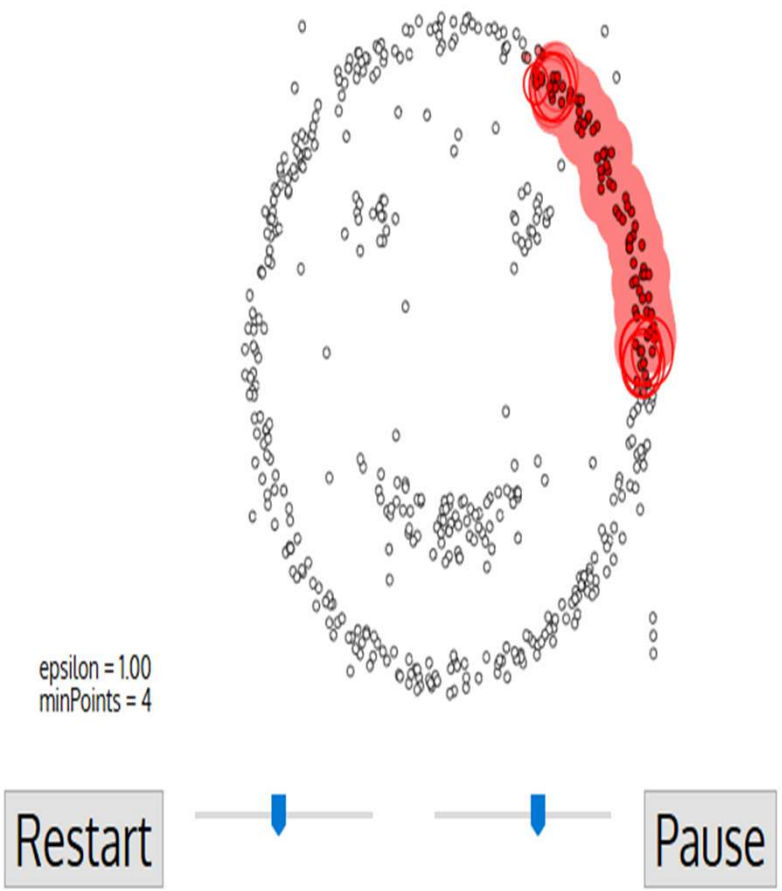


DBSCAN- Density Based Spatial Clustering of Applications with Noise

- It groups densely grouped data points into a single cluster. It can identify clusters in large spatial datasets by looking at the local density of the data points.



- **Core** — This is a point that has at least *m* points within distance *n* from itself.
- **Border** — This is a point that has at least one Core point at a distance *n*.
- **Noise** — This is a point that is neither a Core nor a Border. And it has less than *m* points within distance *n* from itself.



Algorithmic steps for DBSCAN clustering

- The algorithm proceeds by arbitrarily picking up a point in the dataset (until all points have been visited).
- If there are at least ‘minPoint’ points within a radius of ‘ε’ to the point, then we consider all these points to be part of the same cluster.
- The clusters are then expanded by recursively repeating the neighborhood calculation for each neighboring point.