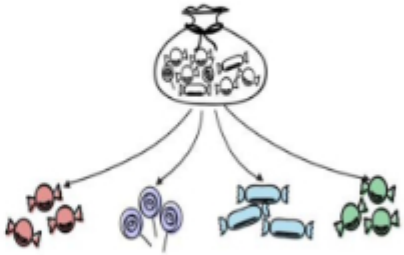
Clustering

Dr. Amit Praseed

What is clustering?

- Clustering is the task of grouping together similar data items in a dataset
- Eg: Similar users are grouped together in a recommendation system
- The class label is often absent in clustering algorithms, which differentiates it from classification



Basic Clustering Techniques

- **Partition Based Clustering**
  - divides the data into k groups such that each group must contain at least one object
  - exclusive cluster separation
- **Hierarchical Clustering**
  - creates a hierarchical decomposition of the given set of data objects
- **Density based Clustering**
  - continue growing a given cluster as long as the density in the "neighborhood" exceeds some threshold
- **Grid Based Clustering**
  - quantizes the object space into a finite number of cells that form a grid structure

k-means Algorithm

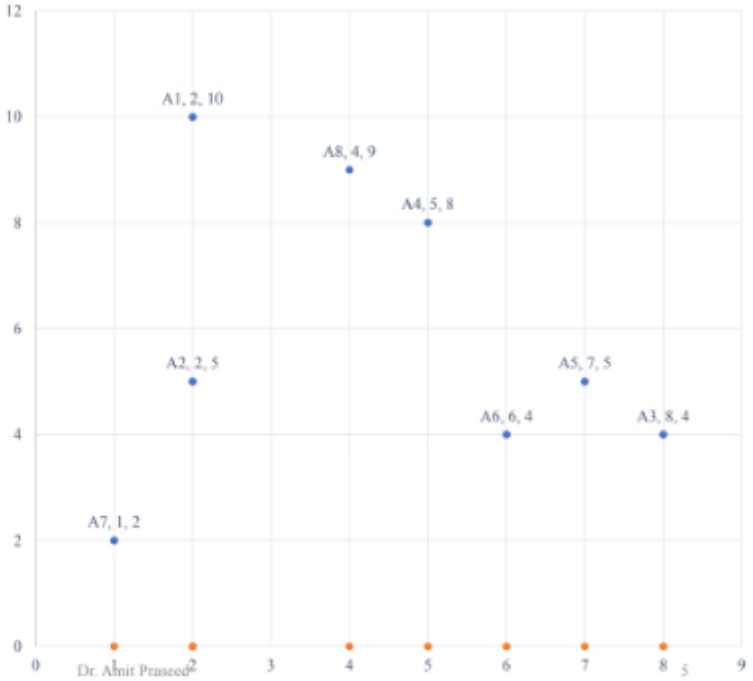
- The centroid of a cluster denotes that cluster
  - For the k-means algorithm, the mean is used to denote the centroid
- Quality of a cluster depends on how similar the items are within a cluster – minimize the within cluster variation

$$E = \sum_{i=1}^{k} \sum_{p \in C_i} dist(p, C_i)$$

- In general, the problem is NP-Hard
- k-means algorithm uses a greedy approach to approximate the process

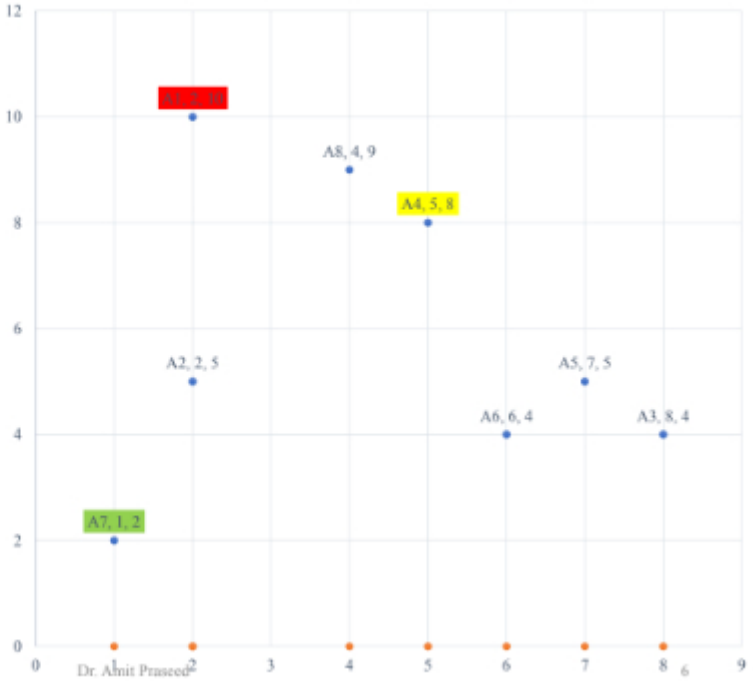
Example

Data Point	X	Y
A1	2	10
A2	2	5
A3	8	4
A4	5	8
A5	7	5
A6	6	4
A7	1	2
A8	4	9



Example

Data Point	X	Y
A1 (Red)	2	10
A2	2	5
A3	8	4
A4 (Yellow)	5	8
A5	7	5
A6	6	4
A7 (Green)	1	2
A8	4	9



Calculate Distance (Eg: Manhattan Distance)

Data Point	Distance from Red Cluster (2,10)	Distance from Yellow Cluster (5,8)	Distance from Green Cluster (1,2)	Cluster
A1 (2,10)	0	5	9	Red
A2 (2,5)	5	6	4	Green
A3 (8,4)	12	7	9	Yellow
A4 (5,8)	5	0	10	Yellow
A5 (7,5)	10	5	9	Yellow
A6 (6,4)	10	5	7	Yellow
A7 (1,2)	9	10	0	Green
A8 (4,9)	3	2	10	Yellow

First Clusters

Data Point	X	Y
A1 (Red)	2	10
A2 (Green)	2	5
A3(Yellow)	8	4
A4 (Yellow)	5	8
A5(Yellow)	7	5
A6(Yellow)	6	4
A7 (Green)	1	2
A8(Yellow)	4	9

Clusters	Centroid
Red	(2,8)
Green	(1.5,3.5)
Yellow	(6,6)

Recompute Clusters

Data Point	Distance from Red Cluster (2,10)	Distance from Yellow Cluster (6,6)	Distance from Green Cluster (1.5,3.5)	Cluster
A1 (2,10)	0	8	7	Red
A2 (2,5)	5	5	2	Green
A3 (8,4)	12	4	7	Yellow
A4 (5,8)	5	3	8	Yellow
A5 (7,5)	10	2	7	Yellow
A6 (6,4)	10	2	5	Yellow
A7 (1,2)	9	9	2	Green
A8 (4,9)	3	5	8	Red

Second Clusters

Data Point	X	Y
A1 (Red)	2	10
A2 (Green)	2	5
A3(Yellow)	8	4
A4 (Yellow)	5	8
A5(Yellow)	7	5
A6(Yellow)	6	4
A7 (Green)	1	2
A8(Red)	4	9

Clusters	Centroid
Red	(3,9.5)
Green	(1.5,3.5)
Yellow	(6.5,5.25)

Recompute Clusters (again ☹)

Data Point	Distance from Red Cluster (3,9.5)	Distance from Yellow Cluster (6.5,5.25)	Distance from Green Cluster (1.5,3.5)	Cluster
A1 (2,10)	1.5	9.25	7	Red
A2 (2,5)	5.5	4.75	2	Green
A3 (8,4)	10.5	2.75	7	Yellow
A4 (5,8)	3.5	4.25	8	Red
A5 (7,5)	8.5	0.75	7	Yellow
A6 (6,4)	8.5	1.75	5	Yellow
A7 (1,2)	9.5	8.75	2	Green
A8 (4,9)	1.5	6.25	8	Red

Third Clusters

Data Point	X	Y
A1 (Red)	2	10
A2 (Green)	2	5
A3(Yellow)	8	4
A4 (Red)	5	8
A5(Yellow)	7	5
A6(Yellow)	6	4
A7 (Green)	1	2
A8(Red)	4	9

Clusters	Centroid
Red	(3.67,9)
Green	(1.5,3.5)
Yellow	(7,4.3)

Recompute Clusters (again ☹ ☹)

Data Point	Distance from Red Cluster (3.67,9)	Distance from Yellow Cluster (7,4.3)	Distance from Green Cluster (1.5,3.5)	Cluster
A1 (2,10)	2.67	10.7	7	Red
A2 (2,5)	5.67	5.7	2	Green
A3 (8,4)	9.33	1.3	7	Yellow
A4 (5,8)	2.33	5.7	8	Red
A5 (7,5)	7.33	0.7	7	Yellow
A6 (6,4)	7.33	1.3	5	Yellow
A7 (1,2)	9.67	8.3	2	Green
A8 (4,9)	0.33	7.7	8	Red

No Cluster Changes this time ☺

Data Point	Distance from Red Cluster (3.67,9)	Distance from Yellow Cluster (7,4.3)	Distance from Green Cluster (1.5,3.5)	Cluster
A1 (2,10)	2.67	10.7	7	Red
A2 (2,5)	5.67	5.7	2	Green
A3 (8,4)	9.33	1.3	7	Yellow
A4 (5,8)	2.33	5.7	8	Red
A5 (7,5)	7.33	0.7	7	Yellow
A6 (6,4)	7.33	1.3	5	Yellow
A7 (1,2)	9.67	8.3	2	Green
A8 (4,9)	0.33	7.7	8	Red

Final Clusters

Data Point	X	Y
A1 (Red)	2	10
A2 (Green)	2	5
A3(Yellow)	8	4
A4 (Red)	5	8
A5(Yellow)	7	5
A6(Yellow)	6	4
A7 (Green)	1	2
A8(Red)	4	9

Clusters	Centroid
Red	(3.67,9)
Green	(1.5,3.5)
Yellow	(7,4.3)

Summary of k-means Algorithm

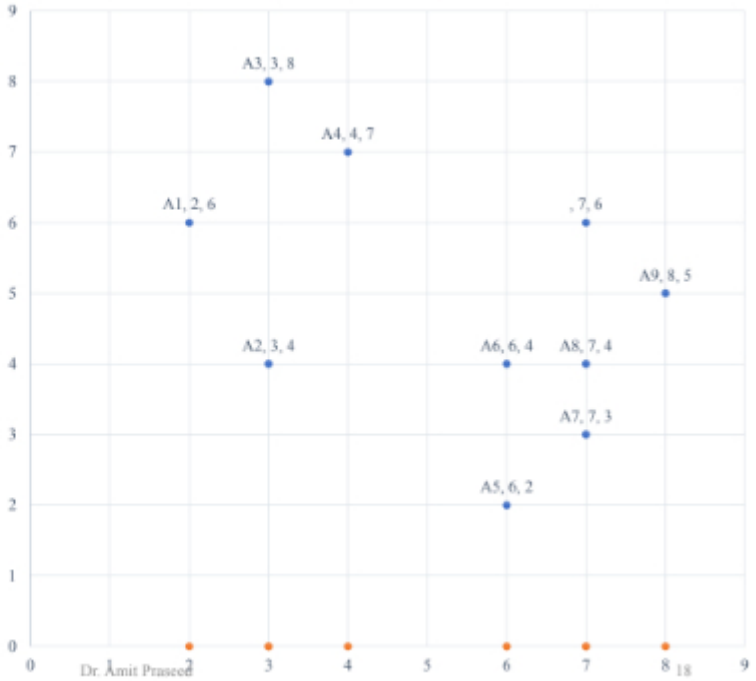
- Not guaranteed to provide a globally optimum solution
- Choosing the optimal k-value is tricky
- Only defined for data types for which mean is defined
  - K-modes is a possible modification
- Can be made more scalable using sampling

k- Medoids Algorithm

- In the k-means algorithm, the centroid is not necessarily one of the data points
  - Sensitive to outliers
- k-medoids algorithm uses a representative element within the group as the “centroid” and computes the clusters based on the medoids
- Partitioning Around Medoids (PAM) algorithm is an example

Example

Data Point	X	Y
A1	2	6
A2	3	4
A3	3	8
A4	4	7
A5	6	2
A6	6	4
A7	7	3
A8	7	4
A9	8	5
A10	7	6



Example

Data Point	X	Y
A1	2	6
A2 (Yellow)	3	4
A3	3	8
A4 (Green)	4	7
A5	6	2
A6	6	4
A7	7	3
A8	7	4
A9	8	5
A10	7	6

Calculate Distance (Eg: Manhattan Distance)

Data Point	Distance from Yellow Cluster (3,4)	Distance from Green Cluster (4,7)	Cluster
A1 (2,6)	3	3	Yellow
A2 (3,4)	0	4	Yellow
A3 (3,8)	4	2	Green
A4 (4,7)	4	0	Green
A5 (6,2)	5	5	Yellow
A6 (6,4)	3	5	Yellow
A7 (7,3)	5	7	Yellow
A8 (7,4)	4	6	Yellow
A9 (8,5)	6	6	Yellow
A10 (7,6)	6	4	Green

In case of clashes, a point is allotted to the Yellow Cluster by default

Example

Data Point	X	Y
A1 (Yellow)	2	6
A2 (Yellow)	3	4
A3 (Green)	3	8
A4 (Green)	4	7
A5 (Yellow)	6	2
A6 (Yellow)	6	4
A7 (Yellow)	7	3
A8 (Yellow)	7	4
A9 (Yellow)	8	5
A10 (Green)	7	6

Compute Absolute Error

Data Point	Distance from Yellow Cluster (3,4)	Distance from Green Cluster (4,7)	Cluster
A1 (2,6)	3	3	Yellow
A2 (3,4) Medoid	0	4	Yellow
A3 (3,8)	4	2	Green
A4 (4,7) Medoid	4	0	Green
A5 (6,2)	5	5	Yellow
A6 (6,4)	3	5	Yellow
A7 (7,3)	5	7	Yellow
A8 (7,4)	4	6	Yellow
A9 (8,5)	6	6	Yellow
A10 (7,6)	6	4	Green

E = (A1-A2) + (A5-A2)+ (A6-A2) + (A7-A2) + (A8-A2)+(A9-A2)
+
(A3-A4)+ (A10-A4)

= (3+4+3+5+4+6)+(2+4)
=31

Example

Data Point	X	Y
A1 (Yellow)	2	6
A2 (Yellow)	3	4
A3 (Green)	3	8
A4 (Green) Medoid	4	7
A5 (Yellow)	6	2
A6 (Yellow) Medoid	6	4
A7 (Yellow)	7	3
A8 (Yellow)	7	4
A9 (Yellow)	8	5
A10 (Green)	7	6

Compute Absolute Error

Data Point	Current Cluster	Distance from Yellow Cluster (6,4)	Distance from Green Cluster (4,7)	Cluster	Error
A1 (2,6)	Yellow	6	3	Green	3
A2 (3,4)	Yellow	3	4	Yellow	3
A3 (3,8)	Green	7	2	Green	2
A4 (4,7) Medoid	Green	5	0	Green	0
A5 (6,2)	Yellow	2	5	Yellow	2
A6 (6,4) Medoid	Yellow	0	5	Yellow	0
A7 (7,3)	Yellow	2	7	Yellow	2
A8 (7,4)	Yellow	1	6	Yellow	1
A9 (8,5)	Yellow	3	6	Yellow	3
A10 (7,6)	Green	3	4	Yellow	3
					19

Example

Data Point	X	Y
A1 (Yellow)	2	6
A2 (Yellow)	3	4
A3 (Green)	3	8
A4 (Green) Medoid	4	7
A5 (Yellow)	6	2
A6 (Yellow) Medoid	6	4
A7 (Yellow)	7	3
A8 (Yellow)	7	4
A9 (Yellow)	8	5
A10 (Green)	7	6

The error reduces, so we use the new set of medoids

PAM Algorithm

- The algorithm starts with a randomly selected set of medoids
- Each point is allocated to a particular cluster based on how close they are to the representative elements
- Randomly select a non-representative element to replace an existing representative element
- If the cost after replacement reduces, the new set of representative elements is retained, else it is discarded
- More robust than k-means
- High complexity – $O(k(n-k)^2)$

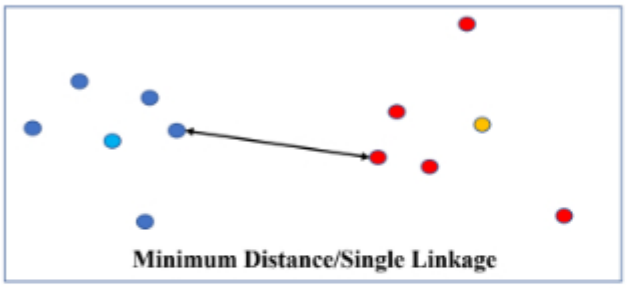
Scalable Versions of PAM

- Clustering LARge Applications (CLARA)
  - Select a random sample from the data points and perform the PAM algorithm
  - Success depends on how well the sample represents the population
- Clustering Large Applications based on RANdomised Search (CLARANS)
  - Confine the set of candidate replacement medoids to a random sample of the data

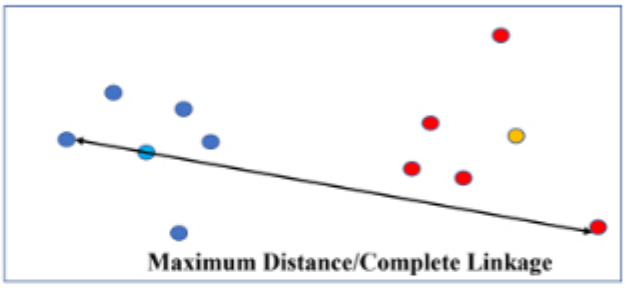
Hierarchical Clustering

- Groups data objects into a hierarchy or “tree” of clusters
- Agglomerative Hierarchical Clustering:
  - Bottom Up
  - Starts with every point in a separate cluster
  - Clusters are merged together based on how “close” they are
  - Finally, you get one “super cluster”
- Divisive Hierarchical Clustering:
  - Top Down
  - Starts with a single “super cluster”
  - Iteratively splits the clusters so that cohesion within the cluster improves
  - Finally every point becomes its own cluster

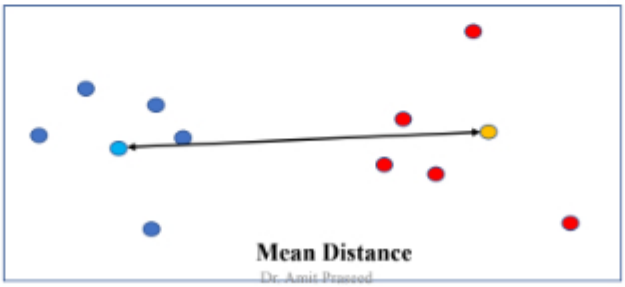
Linkage Measures between Clusters



Minimum Distance/Single Linkage

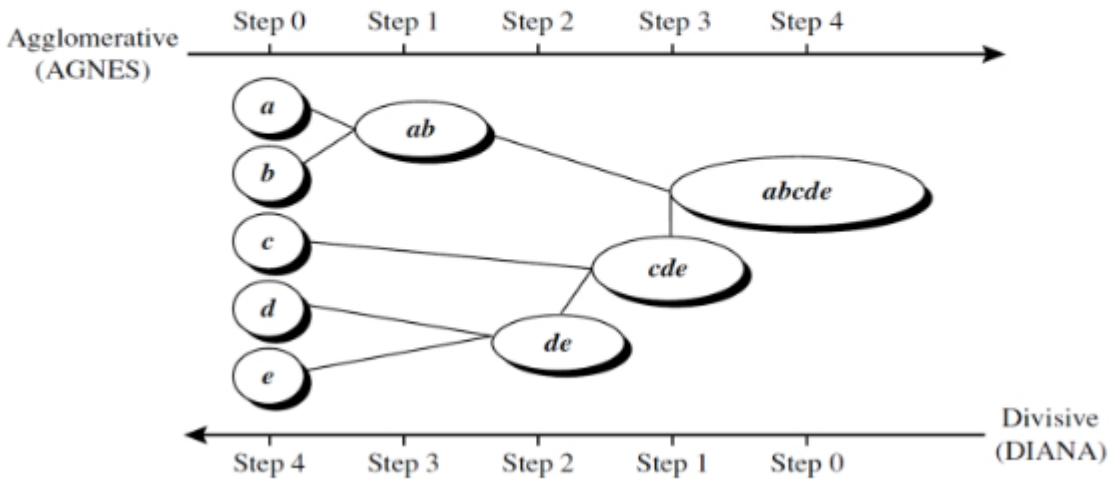


Maximum Distance/Complete Linkage



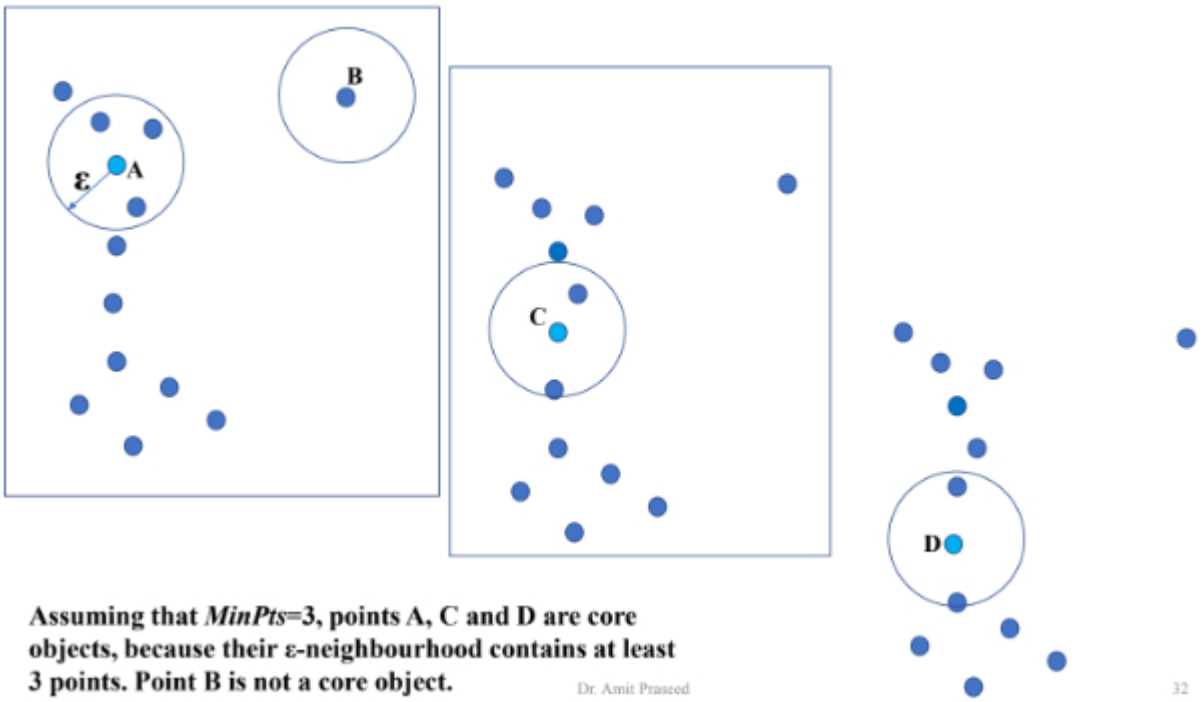
Mean Distance

Dendrogram Representation



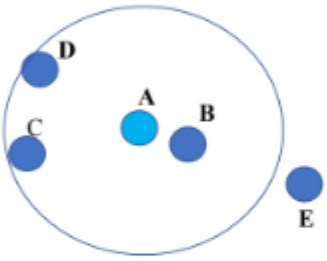
Density Based Clustering

- The density of an object *o* can be measured by the number of objects close to *o*.
- DBSCAN (Density-Based Spatial Clustering of Applications with Noise) finds core objects, that is, objects that have dense neighborhoods.
  - It connects core objects and their neighborhoods to form dense regions as clusters.
- A user-specified parameter ε is used to specify the radius of a neighborhood we consider for every object.
- An object is a core object if the ε - neighborhood of the object contains at least *MinPts* objects.

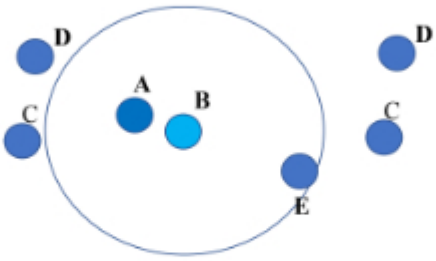


Assuming that *MinPts*=3, points A, C and D are core objects, because their ε-neighbourhood contains at least 3 points. Point B is not a core object.

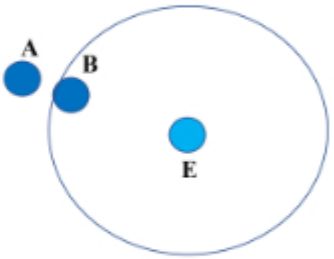
Core, Border and Noise Objects



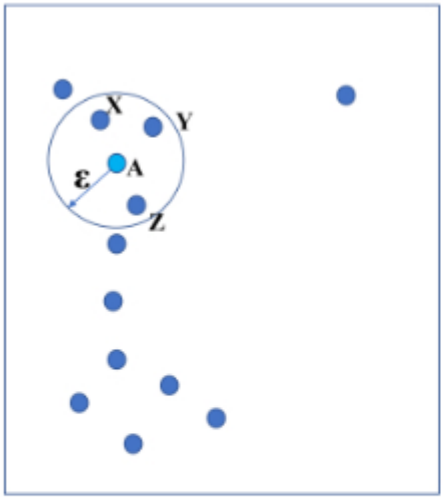
Assuming that *MinPts*=4, object A is a core object because its ε-neighbourhood contains 5 objects



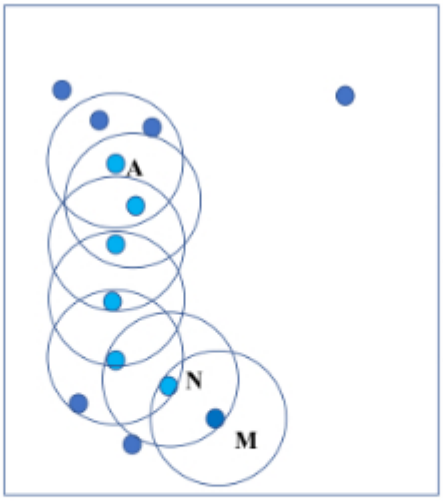
Assuming that *MinPts*=4, object B is called a Border object, because it lies in the neighbourhood of a core object (A), but is itself not a core object



Assuming that *MinPts*=4, object E is called a Noise Object, because it is neither a core object, nor a border object



X, Y and Z are said to be direct density reachable from A

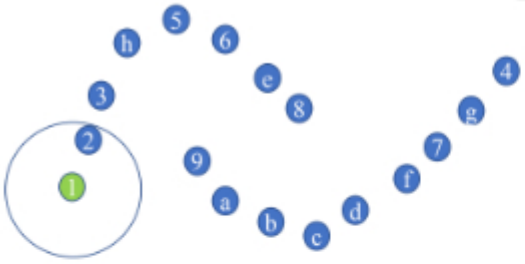


M and N are said to be density reachable from A

Here N is density reachable from A and A is density reachable from N.
Hence, we say that A and N are density connected.

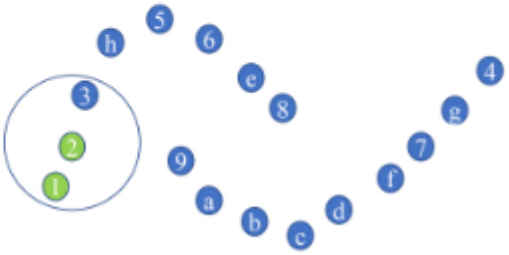
We cannot say the same for A and M.

Let MinPts=2

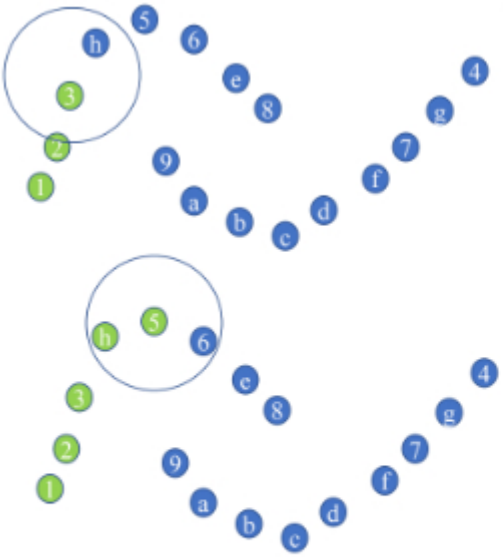


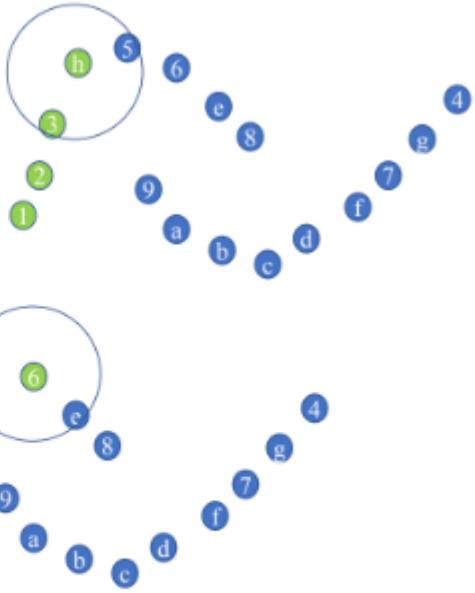
For every unvisited node in N, assign it to the current cluster (green) and add its neighbours to N if it is a core node

Select a random, unvisited object. If it is a core object assign it to a cluster, say green and add all of its neighbours into a candidate set N. Otherwise mark it as a noise node.



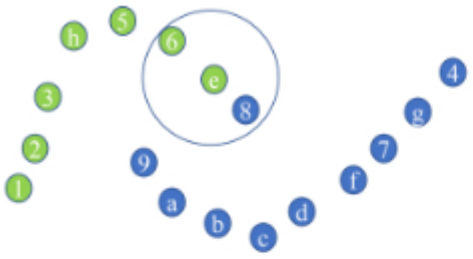
Let MinPts=2

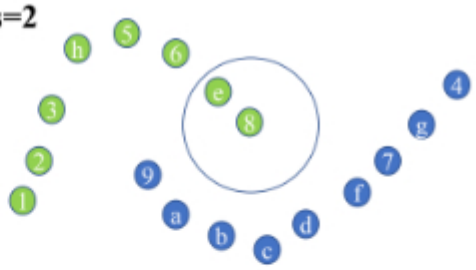

h
5
6
e
8
4
g
3
7
2
f
1
9
a
b
d
c


h
5
6
e
8
4
g
3
7
2
f
1
9
a
b
d
c

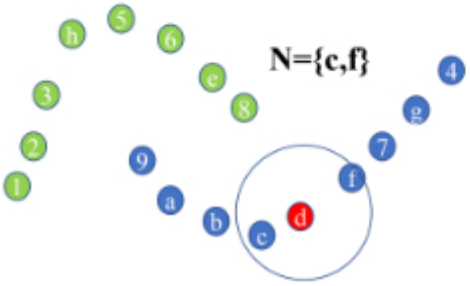
h
5
6
e
8
4
g
3
7
2
f
1
9
a
b
d
c

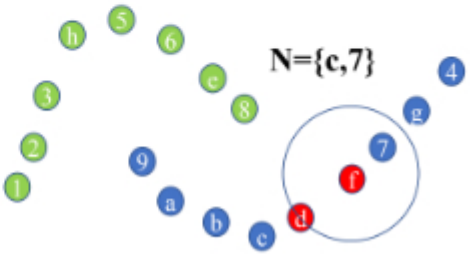
h
5
6
e
8
4
g
3
7
2
f
1
9
a
b
d
c

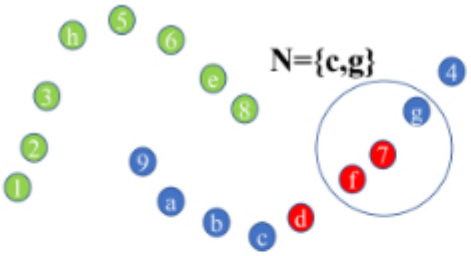


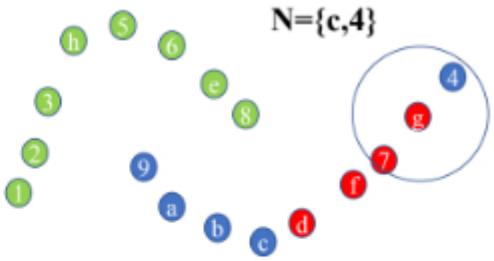


At this point, the set N becomes empty. So DBSCAN picks another unvisited node and adds it to a new cluster.









Finally, when no more unvisited nodes are left…

Summary of DBSCAN Algorithm

- Capable of detecting non-spherical clusters
- Drawbacks:
  - Algorithm is sensitive to the value of ε and MinPts, which are difficult to estimate
  - In real world scenarios, use of a global density value may not yield good results

Clustering in the Presence of Query Conditions

- One of the common applications of clustering is spatial data mining
- It is common to encounter problems where the required clusters have conditions attached to them
  - Eg: Select the maximal regions that have at least 100 houses per unit area and at least 70% of the house prices are above $400K and with total area at least100 units with 90% confidence
- Every query requires a separate clustering operation
  - Each clustering operation depends on the number of objects *n* in the dataset
  - Extensive computation required *per query*

Grid Based Clustering

- Grid-based clustering method takes a space-driven approach
  - Partition the embedding space into cells
  - Independent of the distribution of the input objects
- Quantizes the object space into a finite number of cells that form a grid structure
- Fast processing time, typically independent of the number of data
  - dependent on only the number of cells in each dimension in the quantized space.
- Eg: STING, CLIQUE

STING Algorithm

- STING (STatistical INformation Grid) is a grid based clustering algorithm for answering queries
  - Eg: Select the maximal regions that have at least 100 houses per unit area and at least 70% of the house prices are above $400K and with total area at least100 units with 90% confidence
- Divides the entire search space hierarchically into cells
- At the bottom layer, metrics such as number of points, mean, standard deviation, min, max etc. are maintained
- The information of higher layers can be computed from the information at lower layers.

STING Algorithm

- Query answering starts at a particular layer.
  - Cells which satisfy the constraints are selected based on confidence interval
  - Processing at the next layer only requires selected cells
  - Proceed till the last layer
- Query independent
- Cost depends on the granularity at the lowest level
- All cluster shapes are isothetic

CLIQUE Algorithm

- CLIQUE (CLustering In QUEst) is a grid based method for finding density based clusters in subspaces
- Uses the Apriori property:
  - A *k*-dimensional cell *c* can have at least *l* points only if every *(k-1)*-dimensional projection of *c* has at least *l* points
- Dense clusters are identified in (k-1) dimension and the candidate clusters for the kth dimension are found similar to apriori algorithm