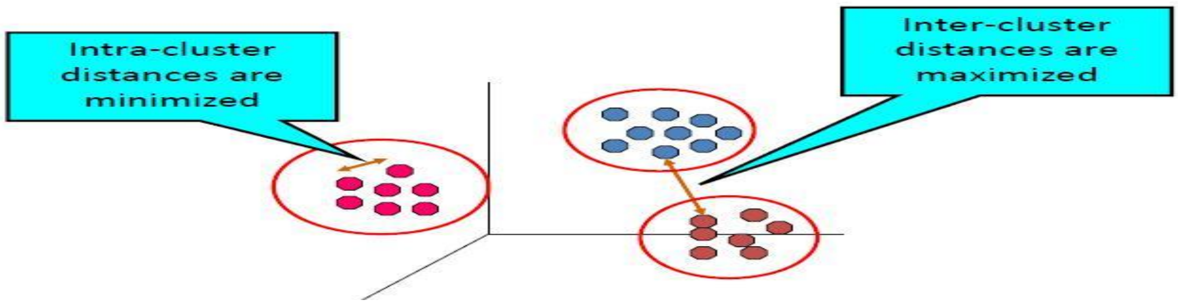
Unit 5: Cluster Analysis

Cluster:

- Clustering is the process of grouping the data into classes or clusters, so that objects within a cluster have high similarity in comparison to one another but are very dissimilar to objects in other clusters.
- **Clustering is the task of dividing the population or data points into a number of groups such that data points in the same groups are more similar to other data points in the same group than those in other groups**
- Let's understand this with an example. Suppose, you are the head of a rental store and wish to understand preferences of your costumers to scale up your business. Is it possible for you to look at details of each costumer and devise a unique business strategy for each one of them? Definitely not. But, what you can do is to cluster all of your costumers into say 10 groups based on their purchasing habits and use a separate strategy for costumers in each of these 10 groups. And this is what we call clustering.
- Clustering is **also called data segmentation** in some applications because clustering partitions large data sets into groups according to their similarity. Clustering can also be **used for outlier detection**, where outliers may be more interesting than common cases. Applications of outlier detection include the detection of credit card fraud and the monitoring of criminal activities in electronic commerce. For example, exceptional cases in credit card transactions, such as very expensive and frequent purchases, may be of interest as possible fraudulent activity.
- In machine learning, clustering is an example of **unsupervised learning.** Unlike classification, clustering and unsupervised learning do not rely on predefined classes and class-labeled training examples. For this reason, clustering is a form of learning by observation, rather than learning by examples.
- Finding groups of objects such that the objects in a group will be similar (or related) to one another and different from (or unrelated to) the objects in other groups.


Intra-cluster distances are minimized
Inter-cluster distances are maximized

Examples of Clustering Applications

- Marketing: Help marketers discover distinct groups in their customer bases, and then use this knowledge to develop targeted marketing programs
- Land use: Identification of areas of similar land use in an earth observation database
- Insurance: Identifying groups of motor insurance policy holders with a high average claim cost
- City-planning: Identifying groups of houses according to their house type, value, and geographical location

What Is Good Clustering?

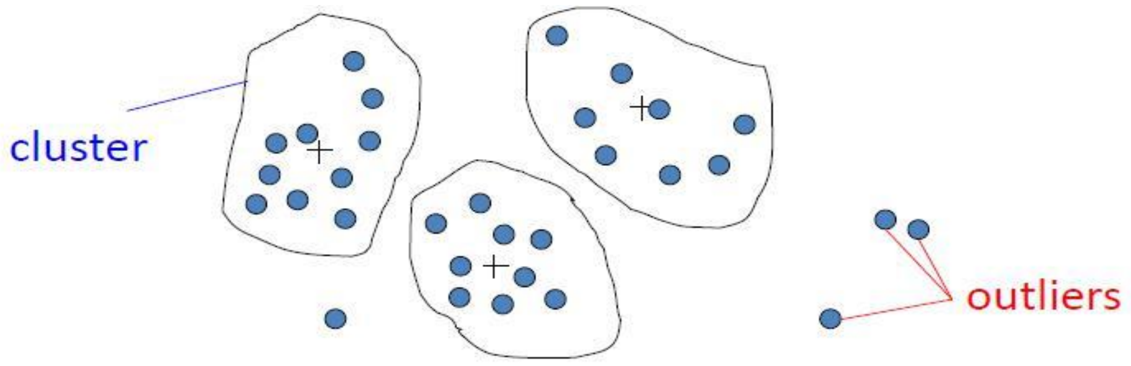
A good clustering method will produce high quality clusters with

- – high intra-class similarity
- – low inter-class similarity

- The quality of a clustering result depends on both the similarity measure used by the method and its implementation.
- The quality of a clustering method is also measured by its ability to discover some or all of the hidden patterns.

Outliers

Outliers are objects that do not belong to any cluster or form clusters of very small cardinality



•In some applications we are interested in discovering outliers, not clusters (outlier analysis)

<u>Types of Clustering</u>

Broadly speaking, clustering can be divided into two subgroups :

- **Hard Clustering**: In hard clustering, each data point either belongs to a cluster completely or not. For example, in the above example each customer is put into one group out of the 10 groups.
- **Soft Clustering**: In soft clustering, instead of putting each data point into a separate cluster, a probability or likelihood of that data point to be in those clusters is assigned. For

example, from the above scenario each costumer is assigned a probability to be in either of 10 clusters of the retail store.

Types of clustering algorithms

Since the task of clustering is subjective, the means that can be used for achieving this goal are plenty. Every methodology follows a different set of rules for defining the 'similarity' among data points. In fact, there are more than 100 clustering algorithms known. But few of the algorithms are used popularly, let's look at them in detail:

- **Partitioning algorithms**: Construct random partitions and then iteratively refine them by some criterion.
- Hierarchical algorithms: Create a hierarchical decomposition of the set of data (or objects) using some criterion.
- Density-based: based on connectivity and density functions
- Grid-based: based on a multiple-level granularity structure
- Model-based: A model is hypothesized for each of the clusters and the idea is to find the best fit of that model to each other

Partitioning algorithms

Given a database of n objects or data tuples, a partitioning method constructs k partitions of the data, where each partition represents a cluster and k <= n. That is, it classifies the data into k groups, which together satisfy the following requirements: (1) each group must contain at least one object, and (2) each object must belong to exactly one group. Notice that the second requirement can be relaxed in some fuzzy partitioning techniques.

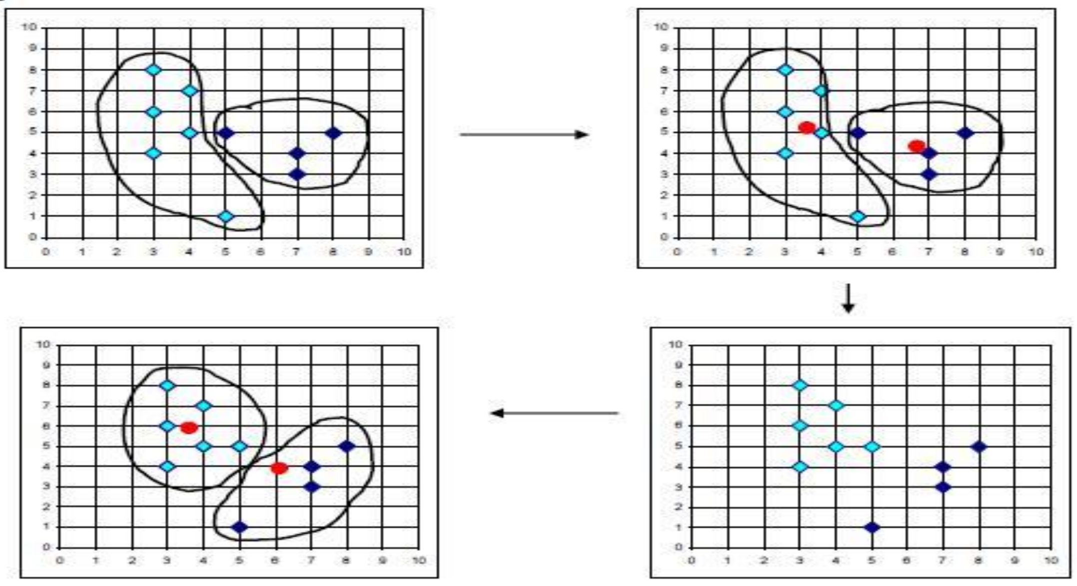
Given k, the number of partitions to construct, a partitioning method creates an initial partitioning. It then uses an iterative relocation technique that attempts to improve the partitioning by moving objects from one group to another. The general criterion of a good partitioning is that objects in the same cluster are "close" or related to each other, whereas objects of different clusters are "far apart" or very different.

The k-means Clustering Method

Given k, the k-means algorithm is implemented in 4 steps:

1. Partition objects into k nonempty subsets
2. Compute seed points as the centroids of the clusters of the current partition. The centroid is the center (mean point) of the cluster.
3. Assign each object to the cluster with the nearest seed point.
4. Go back to Step 2, stop when no more new assignment.

Example



K-Means example

2, 3, 6, 8, 9, 12, 15, 18, 22 –break into 3 clusters

Cluster 1 = 2, 8, 15 => mean = 8.3

Cluster 2 = 3, 9, 18 => mean = 10

Cluster 3 = 6, 12, 22 => mean = 13.3

- Re-assign

Cluster 1 = 2, 3, 6, 8, 9 => mean = 5.6

Cluster 2 = > mean = 0

Cluster 3 = 12, 15, 18, 22 => mean = 16.75

- Re-assign

Cluster 1 =3, 6, 8, 9 => mean = 6.5

Cluster 2 =2 => mean = 2

Cluster 3 = 12, 15, 18, 22 => mean = 16.75

- Re-assign

- Re-assign

Cluster 1 =6, 8, 9 =>mean = 7.6

Cluster 2 =2, 3 =>mean = 2.5

Cluster 3 =12, 15, 18, 22 =>mean = 16.75

•No change, so we’re done

K-Means example –different starting order

•2, 3, 6, 8, 9, 12, 15, 18, 22 –break into 3 clusters

–Cluster 1 -2, 12, 18 –mean = 10.6

–Cluster 2 -6, 9, 22 –mean = 12.3

–Cluster 3 –3, 8, 15 –mean = 8.6

•Re-assign

–Cluster 1 -mean = 0

–Cluster 2 –12, 15, 18, 22 -mean = 16.75

–Cluster 3 –2, 3, 6, 8, 9 –mean = 5.6

•Re-assign

–Cluster 1 –2 –mean = 2

–Cluster 2 –12, 15, 18, 22 –mean = 16.75

–Cluster 3 = 3, 6, 8, 9 –mean = 6.5

•Re-assign

–Cluster 1 –2, 3 –mean = 2.5

–Cluster 2 –12, 15, 18, 22 –mean = 16.75

–Cluster 3 –6, 8, 9 –mean = 7.6

•Re-assign

–Cluster 1 –2, 3 –mean = 2.5

–Cluster 2 –12, 15, 18, 22 -mean = 16.75

–Cluster 3 –6, 8, 9 –mean = 7.6

•No change, so we’re done

Strength

–Relatively efficient: O(tkn), where n is no. objects, k is no clusters, and t is no iterations. Normally, k, t<< n.

Weaknesses

–Applicable only when mean is defined, then what about categorical data?

–Need to specify k, the number of clusters, in advance

–Unable to handle noisy data and outliers

Hierarchical methods:

A hierarchical method creates a hierarchical decomposition of the given set of data objects. A hierarchical method can be **classified as being either agglomerative or divisive**, based on how the hierarchical decomposition is formed. The **agglomerative approach, also called the bottom-up approach, starts with each object forming a separate group**. **It successively merges the objects or groups that are close to one another, until all of the groups are merged into one** (the topmost level of the hierarchy), or until a termination condition holds. The **divisive approach, also called the top-down approach, starts with all of the objects in the same cluster. In each successive iteration, a cluster is split up into smaller clusters, until eventually each object is in one cluster**, or until a termination condition holds. Hierarchical methods suffer from the fact that once a step (merge or split) is done, it can never be undone. This rigidity is useful in that it leads to smaller computation costs by not having to worry about a combinatorial number of different choices.

Agglomerative versus divisive hierarchical clustering. Figure below shows the application of AGNES (AGglomerative NESting), an agglomerative hierarchical clustering method, and DIANA (DIvisive ANAlysis), a divisive hierarchical clustering method, to a data set of five objects, {a, b, c, d, e}. Initially, AGNES places each object into a cluster of its own. The clusters are then merged step-by-step according to some criterion. For example, clusters C1 and C2 may be merged if an object in C1 and an object in C2 form the minimum Euclidean distance between any two objects from different clusters. This is a single-linkage approach in that each cluster is represented by all of the objects in the cluster, and the similarity between two clusters is measured by the similarity of the closest pair of data points belonging to different clusters. The cluster merging process repeats until all of the objects are eventually merged to form one cluster.

In DIANA, all of the objects are used to form one initial cluster. The cluster is split according to some principle, such as the maximum Euclidean distance between the closest neighboring objects in the cluster. The cluster splitting process repeats until, eventually, each new cluster contains only a single object.

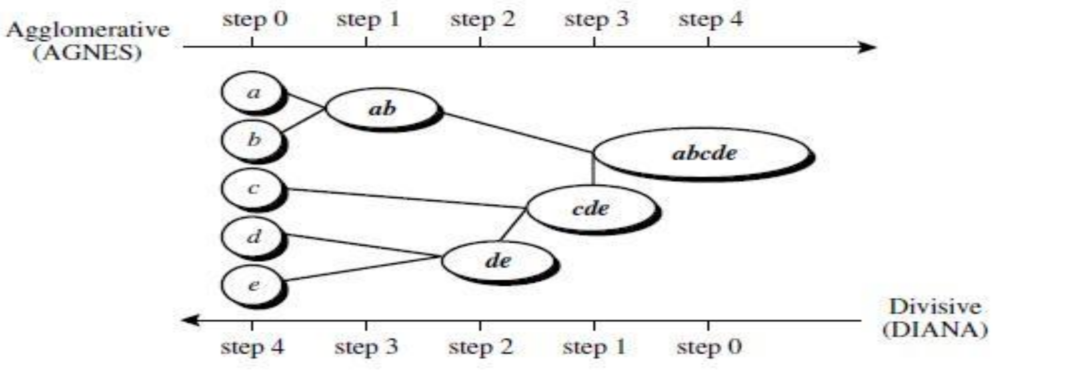


Figure (a): Agglomerative and divisive hierarchical clustering on data objects {a, b, c, d, e}.

In either agglomerative or divisive hierarchical clustering, the user can specify the desired number of clusters as a termination condition.

A tree structure called a dendrogram is commonly used to represent the process of hierarchical clustering. It shows how objects are grouped together step by step. Figure below shows a dendrogram for the five objects presented in Figure a, where l = 0 shows the five objects as singleton clusters at level 0. At l = 1, objects a and b are grouped together to form the first cluster, and they stay together at all subsequent levels. We can also use a vertical axis to show the similarity scale between clusters. For example, when the similarity of two groups of objects, {a, b} and {c, d, e}, is roughly 0.16, they are merged together to form a single cluster.

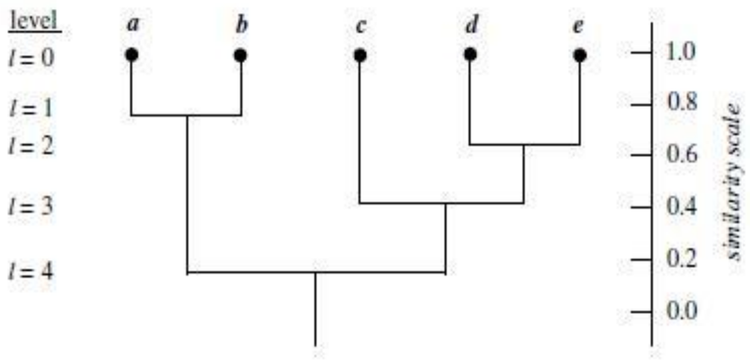
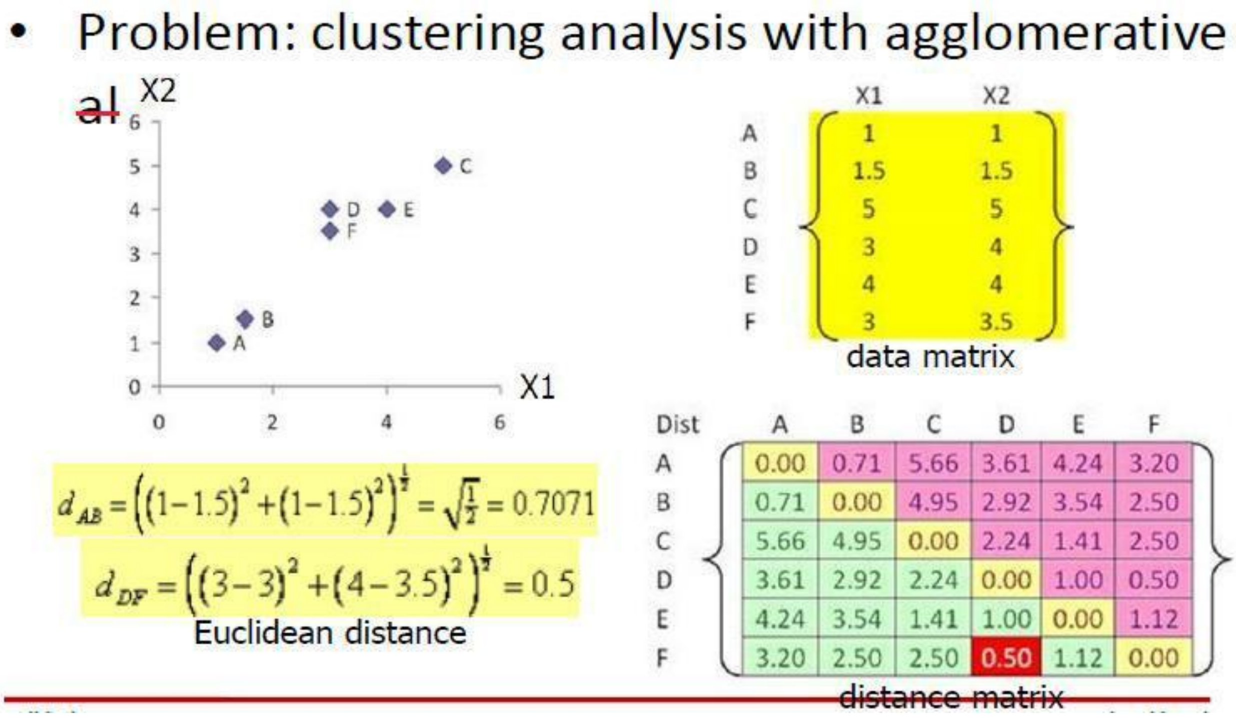


Figure: Dendrogram representation for hierarchical clustering of data objects {a, b, c, d, e}.

Example and Demo

Problem: clustering analysis with agglomerative



• Problem: clustering analysis with agglomerative ~~al~~

	X1	X2
A	1	1
B	1.5	1.5
C	5	5
D	3	4
E	4	4
F	3	3.5

data matrix

$$d_{AB} = \left((1-1.5)^2 + (1-1.5)^2\right)^{\frac{1}{2}} = \sqrt{\tfrac{1}{2}} = 0.7071$$

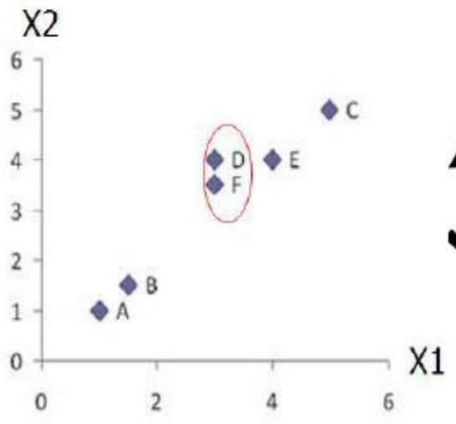
$$d_{DF} = \left((3-3)^2 + (4-3.5)^2\right)^{\frac{1}{2}} = 0.5$$

Euclidean distance

Dist	A	B	C	D	E	F
A	0.00	0.71	5.66	3.61	4.24	3.20
B	0.71	0.00	4.95	2.92	3.54	2.50
C	5.66	4.95	0.00	2.24	1.41	2.50
D	3.61	2.92	2.24	0.00	1.00	0.50
E	4.24	3.54	1.41	1.00	0.00	1.12
F	3.20	2.50	2.50	0.50	1.12	0.00

distance matrix

- Merge two closest clusters



Dist	A	B	C	D	E	F
A	0.00	0.71	5.66	3.61	4.24	3.20
B	0.71	0.00	4.95	2.92	3.54	2.50
C	5.66	4.95	0.00	2.24	1.41	2.50
D	3.61	2.92	2.24	0.00	1.00	0.50
E	4.24	3.54	1.41	1.00	0.00	1.12
F	3.20	2.50	2.50	0.50	1.12	0.00

Dist	A	B	C	D, F	E
A	0.00	0.71	5.66	?	4.24
B	0.71	0.00	4.95	?	3.54
C	5.66	4.95	0.00	?	1.41
D, F	?	?	?	0.00	?
E	4.24	3.54	1.41	?	0.00

- Update distance matrix

Dist	A	B	C	D	E	F
A	0.00	0.71	5.66	3.61	4.24	3.20
B	0.71	0.00	4.95	2.92	3.54	2.50
C	5.66	4.95	0.00	2.24	1.41	2.50
D	3.61	2.92	2.24	0.00	1.00	0.50
E	4.24	3.54	1.41	1.00	0.00	1.12
F	3.20	2.50	2.50	0.50	1.12	0.00

$$d_{(D,F)\mapsto A} = \min(d_{DA}, d_{FA}) = \min(3.61, 3.20) = 3.20$$

$$d_{(D,F)\mapsto B} = \min(d_{DB}, d_{FB}) = \min(2.92, 2.50) = 2.50$$

$$d_{(D,F)\mapsto C} = \min(d_{DC}, d_{FC}) = \min(2.24, 2.50) = 2.24$$

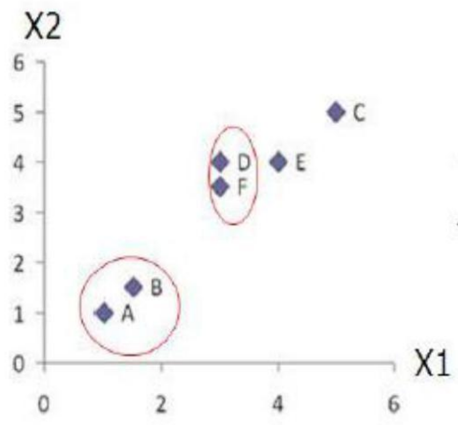
$$d_{E\to(D,F)} = \min(d_{ED}, d_{EF}) = \min(1.00, 1.12) = 1.00$$

Dist	A	B	C	D, F	E
A	0.00	0.71	5.66	?	4.24
B	0.71	0.00	4.95	?	3.54
C	5.66	4.95	0.00	?	1.41
D, F	?	?	?	0.00	?
E	4.24	3.54	1.41	?	0.00

Min Distance (Single Linkage)

Dist	A	B	C	D, F	E
A	0.00	0.71	5.66	3.20	4.24
B	0.71	0.00	4.95	2.50	3.54
C	5.66	4.95	0.00	2.24	1.41
D, F	**3.20**	**2.50**	**2.24**	0.00	1.00
E	4.24	3.54	1.41	**1.00**	0.00

- Merge two closest clusters



Min Distance (Single Linkage)

Dist	A	B	C	D, F	E
A	0.00	0.71	5.66	3.20	4.24
B	0.71	0.00	4.95	2.50	3.54
C	5.66	4.95	0.00	2.24	1.41
D, F	**3.20**	**2.50**	**2.24**	0.00	1.00
E	4.24	3.54	1.41	**1.00**	0.00

Dist	A,B	C	(D, F)	E
A,B	0	?	?	?
C	**?**	0	2.24	1.41
(D, F)	**?**	2.24	0	1.00
E	**?**	1.41	1.00	0

- Update distance matrix

Min Distance (Single Linkage)

Dist	A	B	C	D, F	E
A	0.00	0.71	5.66	3.20	4.24
B	0.71	0.00	4.95	2.50	3.54
C	5.66	4.95	0.00	2.24	1.41
D, F	**3.20**	**2.50**	**2.24**	0.00	1.00
E	4.24	3.54	1.41	**1.00**	0.00

$$d_{C\to(A,B)} = \min(d_{CA}, d_{CB}) = \min(5.66, 4.95) = 4.95$$

$$d_{(D,F)\to(A,B)} = \min(d_{DA}, d_{DB}, d_{FA}, d_{FB}) = \min(3.61, 2.92, 3.20, 2.50) = 2.50$$

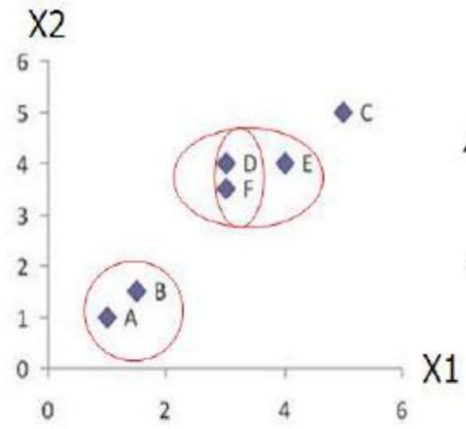
$$d_{E\to(A,B)} = \min(d_{EA}, d_{EB}) = \min(4.24, 3.54) = 3.54$$

Dist	A,B	C	(D, F)	E
A,B	0	?	?	?
C	**?**	0	2.24	1.41
(D, F)	**?**	2.24	0	1.00
E	**?**	1.41	1.00	0

Min Distance (Single Linkage)

Dist	A,B	C	(D, F)	E
A,B	0	4.95	2.50	3.54
C	**4.95**	0	2.24	1.41
(D, F)	**2.50**	2.24	0	1.00
E	**3.54**	1.41	1.00	0

- Merge two closest clusters/update distance matrix



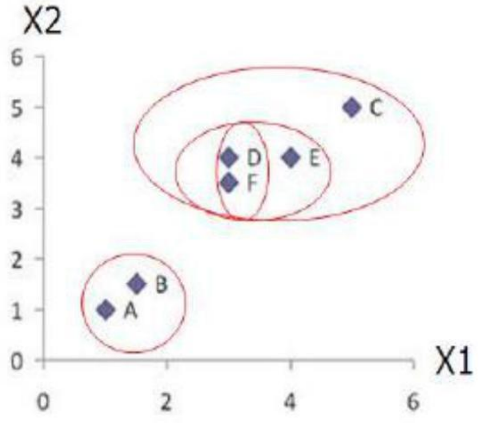
Min Distance (Single Linkage)

Dist	A,B	C	(D, F)	E
A,B	0	4.95	2.50	3.54
C	**4.95**	0	2.24	1.41
(D, F)	**2.50**	2.24	0	1.00
E	**3.54**	1.41	1.00	0

Min Distance (Single Linkage)

Dist	(A,B)	C	(D, F), E
(A,B)	0.00	4.95	2.50
C	4.95	0.00	1.41
(D, F), E	**2.50**	**1.41**	0.00

- Merge two closest clusters/update distance matrix



Min Distance (Single Linkage)

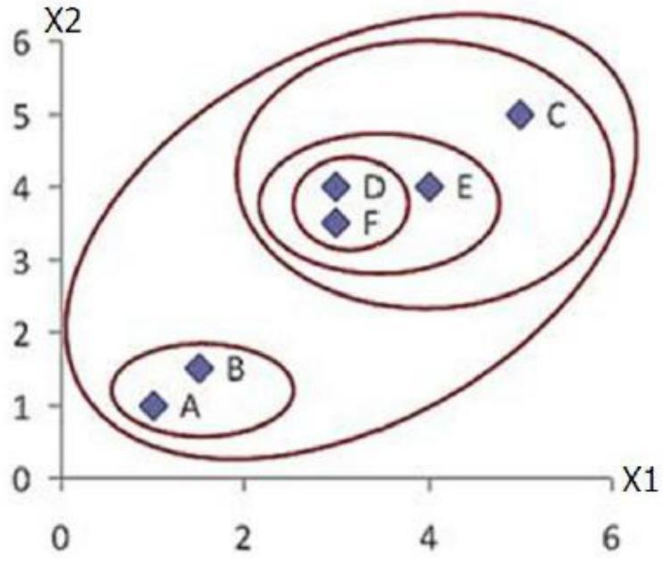
Dist	(A,B)	C	(D, F), E
(A,B)	0.00	4.95	2.50
C	4.95	0.00	1.41
(D, F), E	**2.50**	**1.41**	0.00

Min Distance (Single Linkage)

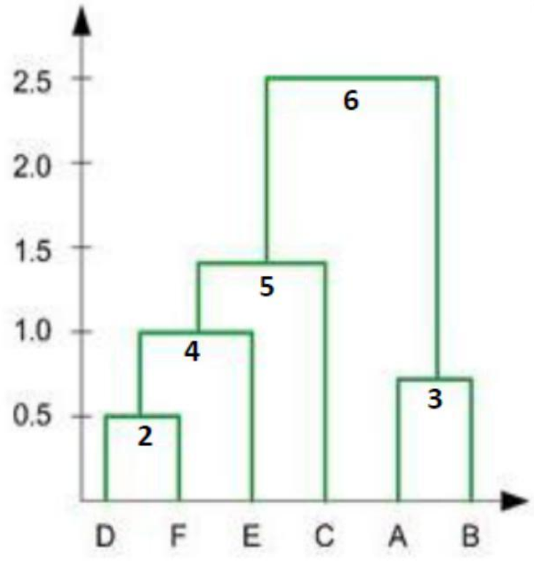
Dist	(A,B)	(D, F), E),C
(A,B)	0.00	2.50
((D, F), E),C	**2.50**	0.00

- Final result (meeting termination condition)

	X1	X2
A	1	1
B	1.5	1.5
C	5	5
D	3	4
E	4	4
F	3	3.5



- Dendrogram tree representation



1. In the beginning we have 6 clusters: A, B, C, D, E and F
2. We merge cluster D and F into cluster (D, F) at distance 0.50
3. We merge cluster A and cluster B into (A, B) at distance 0.71
4. We merge cluster E and (D, F) into ((D, F), E) at distance 1.00
5. We merge cluster ((D, F), E) and C into (((D, F), E), C) at distance 1.41
6. We merge cluster (((D, F), E), C) and (A, B) into ((((D, F), E), C), (A, B)) at distance 2.50
7. The last cluster contain all the objects, thus conclude the computation

Density-based methods:

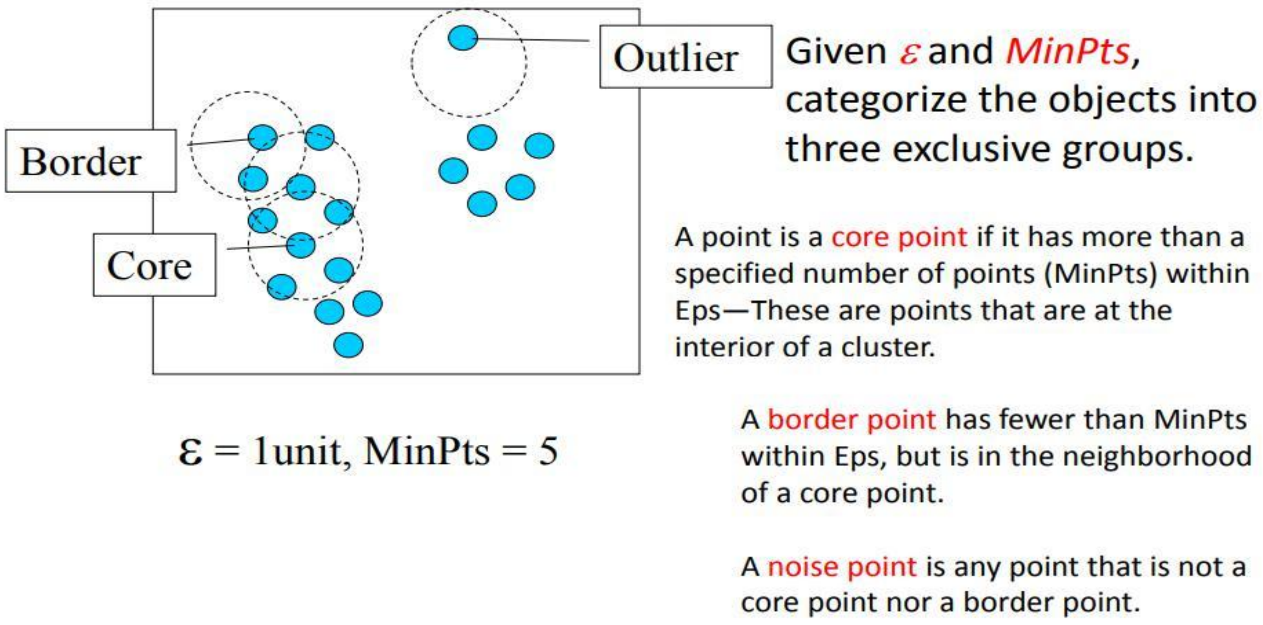
Most partitioning methods cluster objects based on the distance between objects. Such methods can find only spherical-shaped clusters and encounter difficulty at discovering clusters of arbitrary shapes. Other clustering methods have been developed based on the notion of density. Their general idea is to continue growing the given cluster as long as the density (number of objects or data points) in the “neighborhood” exceeds some threshold; that is, for each data point within a given cluster, the neighborhood of a given radius has to contain at least a minimum number of points. Such a method can be used to filter out noise (outliers) and **discover clusters of arbitrary shape.**

DBSCAN and its extension, OPTICS, are typical density-based methods that grow clusters according to a density-based connectivity analysis.

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

DBSCAN (Density-Based Spatial Clustering of Applications with Noise) is a density based clustering algorithm. The algorithm grows regions with sufficiently high density into clusters and discovers clusters of arbitrary shape in spatial databases with noise. It defines a cluster as a maximal set of density-connected points.

- The neighborhood within a radius e of a given object is called the e-neighborhood of the object.
- If the e-neighborhood of an object contains at least a minimum number, MinPts, of objects, then the object is called a core object.



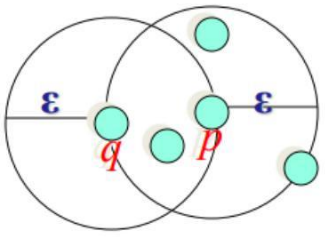
Given ε and *MinPts*, categorize the objects into three exclusive groups.

A point is a core point if it has more than a specified number of points (MinPts) within Eps—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 nor a border point.

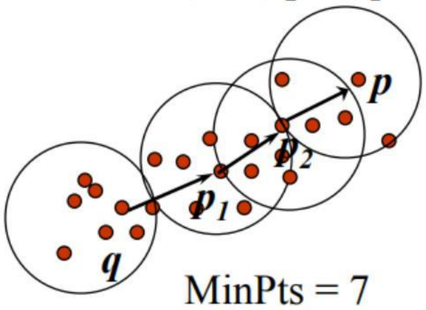
- Directly density-reachable
  - An object *q* is directly density-reachable from object *p* if *p* is a core object and *q* is in p's ε-neighborhood.



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

MinPts = 4

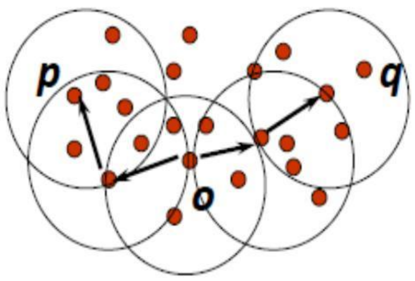
- Density-Reachable (directly and indirectly):
  - A point *p* is directly density-reachable from p_2
  - p_2 is directly density-reachable from p_1
  - p_1 is directly density-reachable from *q*
  - $p \leftarrow p_2 \leftarrow p_1 \leftarrow q$ form a chain



MinPts = 7

- *p* is (indirectly) density-reachable from *q*
- *q* is not density-reachable from *p*

- Density-connected
  - A point *p* is density-connected to a point *q* wrt. *Eps*, *MinPts* if there is a point *o* such that both, *p* and *q* are density-reachable from *o* wrt. *Eps* and *MinPts*.



DBSCAN: The Algorithm

–Arbitrary select a point p

–Retrieve all points density-reachable from p wrt Eps and MinPts.

–If pis a core point, a cluster is formed.

–If p is a border point, no points are density-reachable from p and DBSCAN visits the next point of the database.

–Continue the process until all of the points have been processed.

"How does DBSCAN find clusters?"

DBSCAN searches for clusters by checking the e-neighborhood of each point in the database. If the e-neighborhood of a point p contains more than MinPts, a new cluster with p as a core object is created. DBSCAN then iteratively collects directly density-reachable objects from these core objects, which may involve the merge of a few density-reachable clusters. The process terminates when no new point can be added to any cluster.

Example : Density-reachability and density connectivity. Consider Figure below for a given e represented by the radius of the circles, and, say, let MinPts = 3. Based on the above definitions:

Of the labeled points, m, p, o, and r are core objects because each is in an e-neighborhood containing at least three points.

q is directly density-reachable fromm.mis directly density-reachable from p and vice versa.

q is (indirectly) density-reachable fromp because q is directly density-reachable from m and m is directly density-reachable from p. However, p is not density-reachable from q because q is not a core object. Similarly, r and s are density-reachable from o, and o is density-reachable from r.

o, r, and s are all density-connected.

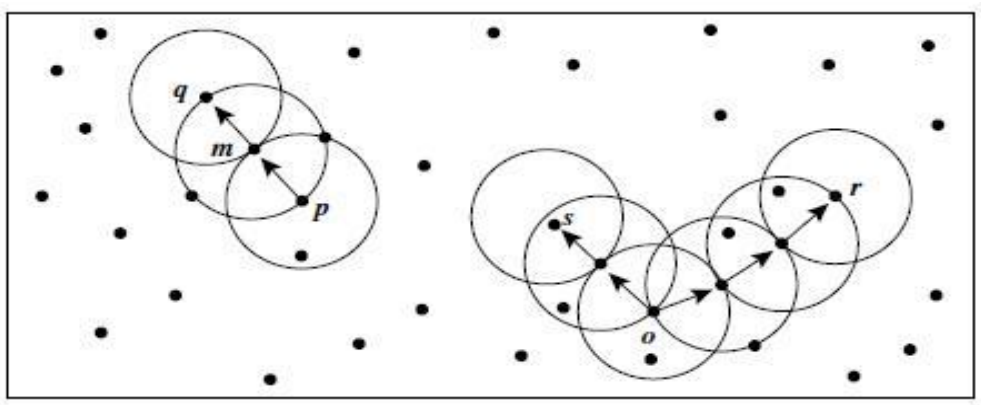


Figure : Density reachability and density connectivity in density-based clustering.