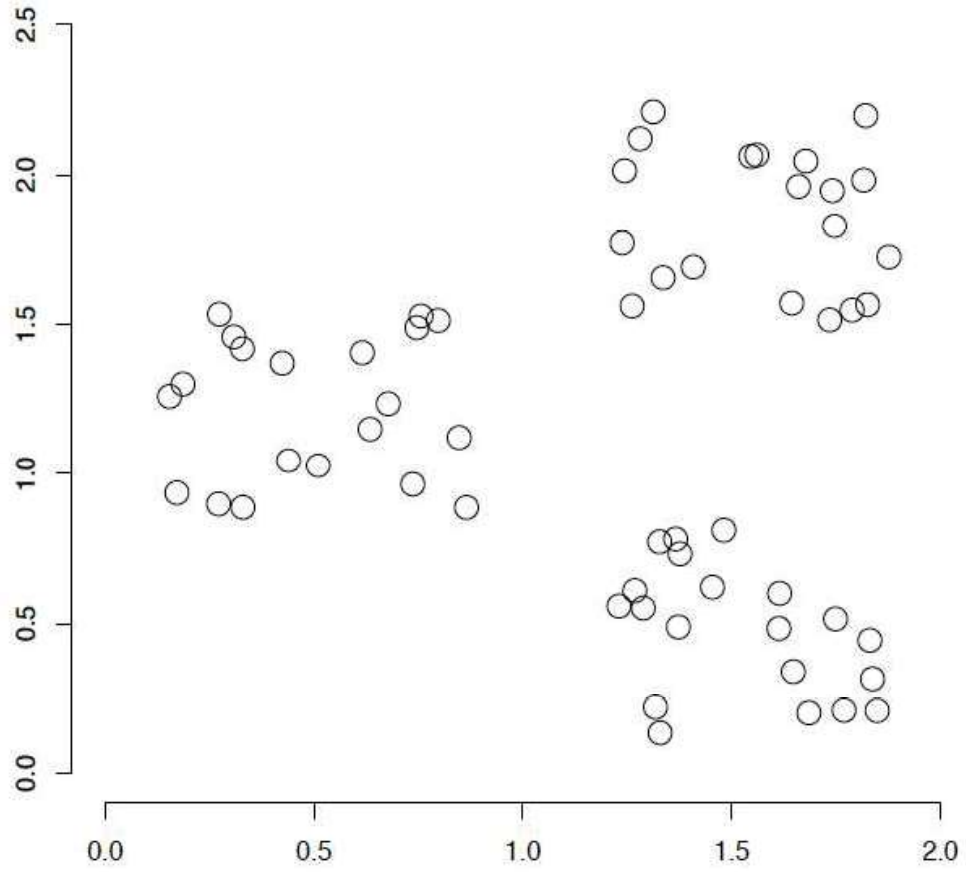
Clustering

- Clustering: the process of grouping a set of objects into classes of similar objects
- *unsupervised learning*
    - Unsupervised learning = learning from raw data, as opposed to supervised data where a classification of examples is given
- Typical applications
    - As a stand-alone tool to get insight into data distribution
    - As a preprocessing step for other algorithms

A data set with clear cluster structure



- How would you design an algorithm for finding the three clusters in this case?

Applications of clustering in IR

- **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
- **Urban planning**: Identifying groups of houses according to their house type, value, and geographical location
- **Seismology**: Observed earth quake epicenters should be clustered along continent faults
- **Biology/genetics**: identify similar entities (organisms/genomes)
- **Content analysis**: Clustering algorithms are used to classify content based on various factors like key terms, sources, and subjects. Many search engines and custom search services use clustering algorithms to classify documents and content according to their categories and search terms.

Good Clustering?

- A good clustering method will produce clusters with
  - High intra-class similarity
  - Low inter-class similarity
- Requires the definition of a similarity measure
- Precise definition of clustering quality is difficult
  - Application-dependent
  - Ultimately subjective

Major Clustering Approaches

- Partitioning approach:
  - Construct various partitions and then evaluate them by some criterion, e.g., minimizing the sum of square errors
  - Typical methods: k-means, k-medoids, CLARANS
- Hierarchical approach:
  - Create a hierarchical decomposition of the set of data (or objects) using some criterion
  - Typical methods: Diana, Agnes, BIRCH, CAMELEON
- Density-based approach:
  - Based on connectivity and density functions
  - Typical methods: DBSACN, OPTICS, DenClue
- Grid-based approach:
  - based on a multiple-level granularity structure
  - Typical methods: STING, WaveCluster, CLIQUE

Major Clustering Approaches

- Model-based:
  - A model is hypothesized for each of the clusters and tries to find the best fit of that model to each other
  - Typical methods: EM, SOM, COBWEB
- Frequent pattern-based:
  - Based on the analysis of frequent patterns
  - Typical methods: p-Cluster
- User-guided or constraint-based:
  - Clustering by considering user-specified or application-specific constraints
  - Typical methods: COD (obstacles), constrained clustering
- Link-based clustering:
  - Objects are often linked together in various ways
  - Massive links can be used to cluster objects: SimRank, LinkClus

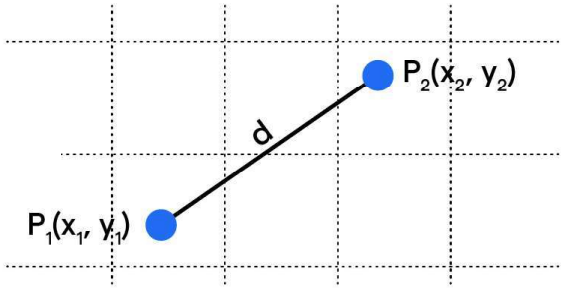
Partitioning Algorithms

- Partitioning method: first, create an initial partitioning, then use iterative relocation techniques to improve the partitioning by moving objects from one group to another.
  - Most partitioning methods are distance-based.
- Given a k, find a partition of k clusters that optimizes the chosen partitioning criterion
  - Global optimal:
    - exhaustively enumerate all partitions
    - computationally prohibitive
  - Heuristic methods: k-means and k-medoids algorithms
  - k-means (MacQueen, 1967): Each cluster is represented by the center of the cluster
  - k-medoids or PAM (Partition around medoids) (Kaufman & Rousseeuw, 1987): Each cluster is represented by one of the objects in the cluster

K-Means

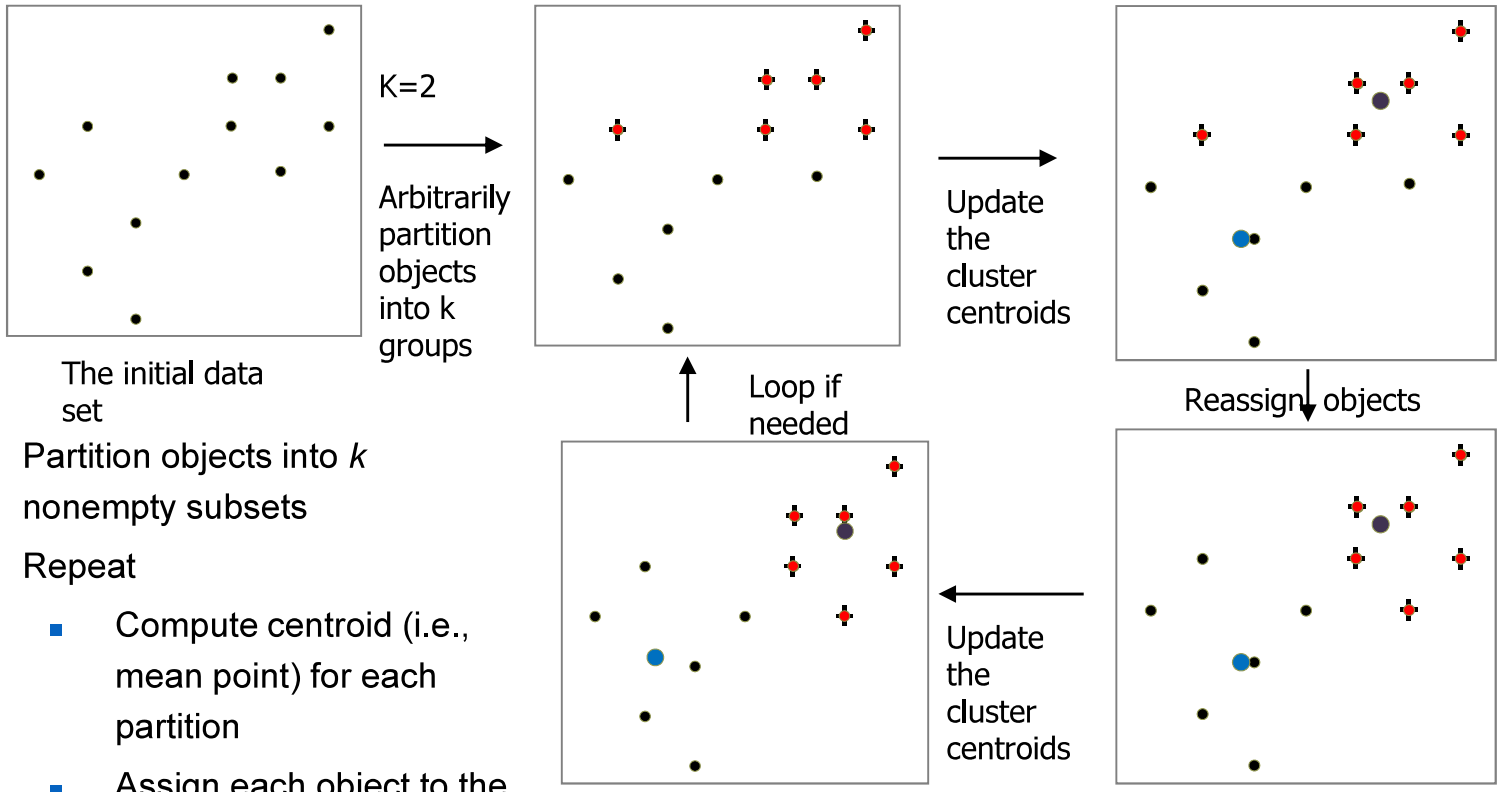
- Given k, the k-means algorithm steps:
  - Partition objects into k nonempty subsets
  - Compute seed points as the centroids of the clusters of the current partitioning (the centroid is the center, i.e., mean point, of the cluster)
  - Assign each object to the cluster with the nearest centroids
    - Euclidean distance
    - Manhattan distance

$$\sqrt{(x_2 - y_1)^2 + (y_2 - y_1)^2}$$



  - Iteratively improves the within-cluster variation
  - Go back to Step 2, stop when the assignment is stable

An Example of K-Means Clustering



- Partition objects into *k* nonempty subsets
- Repeat
  - Compute centroid (i.e., mean point) for each partition
  - Assign each object to the cluster of its nearest centroid
- Until no change

An Example of K-Means Clustering

Point	Coordinates
A1	(2,10)
A2	(2,6)
A3	(11,11)
A4	(6,9)
A5	(6,4)
A6	(1,2)
A7	(5,10)
A8	(4,9)
A9	(10,12)
A10	(7,5)
A11	(9,11)
A12	(4,6)
A13	(3,10)
A14	(3,8)
A15	(6,11)

1. K = 3 random entries from the dataset and use them as centroids.
   Let us consider A2 (2,6), A7 (5,10), and A15 (6,11) as the centroids of the initial clusters.
2. Distance of each entry
3. Assign data points to cluster
4. Calculate the new centroid of the clusters.
5. Stop If the newly created centroids are the same as the centroids in the previous iteration.

K-Means

Point	Coordinates
A1	(2,10)
A2	(2,6)
A3	(11,11)
A4	(6,9)
A5	(6,4)
A6	(1,2)
A7	(5,10)
A8	(4,9)
A9	(10,12)
A10	(7,5)
A11	(9,11)
A12	(4,6)
A13	(3,10)
A14	(3,8)
A15	(6,11)

	Distance from		
Coordinates	C1(2,6)	C2(5,7)	C3(6,11)
(2,10)			
(2,6)			
(11,11)			
(6,9)			
(6,4)			
(1,2)			
(5,10)			
(4,9)			
(10,12)			
(7,5)			
(9,11)			
(4,6)			
(3,10)			
(3,8)			
(6,11)			

K-Means

Point	Coordinates
A1	(2,10)
A2	(2,6)
A3	(11,11)
A4	(6,9)
A5	(6,4)
A6	(1,2)
A7	(5,10)
A8	(4,9)
A9	(10,12)
A10	(7,5)
A11	(9,11)
A12	(4,6)
A13	(3,10)
A14	(3,8)
A15	(6,11)

	Distance from		
Coordinates	**C1(2,6)**	**C2(5,7)**	**C3(6,11)**
(2,10)	4.00	3.00	4.12
(2,6)	0.00	5.00	6.40
(11,11)	10.30	6.08	5.00
(6,9)	5.00	1.41	2.00
(6,4)	4.47	6.08	7.00
(1,2)	4.12	8.94	10.30
(5,10)	5.00	0.00	1.41
(4,9)	3.61	1.41	2.83
(10,12)	10.00	5.39	4.12
(7,5)	5.10	5.39	6.08
(9,11)	8.60	4.12	3.00
(4,6)	2.00	4.12	5.39
(3,10)	4.12	2.00	3.16
(3,8)	2.24	2.83	4.24
(6,11)	6.40	1.41	0.00

Assign Cluster?

K-Means

Point	Coordinates
A1	(2,10)
A2	(2,6)
A3	(11,11)
A4	(6,9)
A5	(6,4)
A6	(1,2)
A7	(5,10)
A8	(4,9)
A9	(10,12)
A10	(7,5)
A11	(9,11)
A12	(4,6)
A13	(3,10)
A14	(3,8)
A15	(6,11)

	Distance from			
Coordinates	**C1(2,6)**	**C2(5,7)**	**C3(6,11)**	**Cluster**
(2,10)	4.00	3.00	4.12	Cluster 2
(2,6)	0.00	5.00	6.40	Cluster 1
(11,11)	10.30	6.08	5.00	Cluster 3
(6,9)	5.00	1.41	2.00	Cluster 2
(6,4)	4.47	6.08	7.00	Cluster 1
(1,2)	4.12	8.94	10.30	Cluster 1
(5,10)	5.00	0.00	1.41	Cluster 2
(4,9)	3.61	1.41	2.83	2
(10,12)	10.00	5.39	4.12	3
(7,5)	5.10	5.39	6.08	1
(9,11)	8.60	4.12	3.00	Cluster 3
(4,6)	2.00	4.12	5.39	Cluster 1
(3,10)	4.12	2.00	3.16	Cluster 2
(3,8)	2.24	2.83	4.24	Cluster 1
(6,11)	6.40	1.41	0.00	Cluster 3

New Cluster Centroid ?

K-Means

Coordinates	Cluster	New Centroid
(2,6)	Cluster 1	(3.8,5,1)
(6,4)		
(1,2)		
(7,5)		
(4,6)		
(3,8)		
(2,10)	Cluster 2	(4,9.6)
(6,9)		
(5,10)		
(4,9)		
(3,10)		
(11,11)	Cluster 3	(9,11.25)
(10,12)		
(9,11)		
(6,11)		

	Distance from			
Coordinates	C1 (3.8,5.1)	C2 (4,9.6)	C3 (9,11.25)	Cluster
(2,10)				
(2,6)				
(11,11)				
(6,9)				
(6,4)				
(1,2)				
(5,10)				
(4,9)				
(10,12)				
(7,5)				
(9,11)				
(4,6)				
(3,10)				
(3,8)				
(6,11)				

K-Means

Coordinates	Cluster	New Centroid
(2,6)	Cluster 1	(3.8,5,1)
(6,4)		
(1,2)		
(7,5)		
(4,6)		
(3,8)		
(2,10)	Cluster 2	(4,9.6)
(6,9)		
(5,10)		
(4,9)		
(3,10)		
(11,11)	Cluster 3	(9,11.25)
(10,12)		
(9,11)		
(6,11)		

	Distance from			
Coordinates	C1 (3.8,5.1)	C2 (4,9.6)	C3 (9,11.25)	Cluster
(2,10)	5.17	2.04	7.11	Cluster 2
(2,6)	2.01	4.12	8.75	Cluster 1
(11,11)	9.24	7.14	2.02	Cluster 3
(6,9)	4.40	2.09	3.75	Cluster 2
(6,4)	2.46	5.95	7.85	Cluster 1
(1,2)	4.25	8.17	12.23	Cluster 1
(5,10)	4.97	1.08	4.19	Cluster 2
(4,9)	3.84	0.60	5.48	Cluster 2
(10,12)	9.20	6.46	1.25	Cluster 3
(7,5)	3.17	5.49	6.56	Cluster 1
(9,11)	7.79	5.19	0.25	Cluster 3
(4,6)	0.85	3.60	7.25	Cluster 1
(3,10)	4.90	1.08	6.13	Cluster 2
(3,8)	2.95	1.89	6.82	Cluster 2
(6,11)	6.22	2.44	3.01	Cluster 2

K-Means

Coordinates	Cluster	New Centroid
(2,6)	Cluster 1	(4,4,6)
(6,4)		
(1,2)		
(7,5)		
(4,6)		
(2,10)	Cluster 2	(4,1,9.5)
(6,9)		
(5,10)		
(4,9)		
(3,10)		
(3,8)		
(6,11)		
(11,11)	Cluster 3	(10,11.3)
(10,12)		
(9,11)		

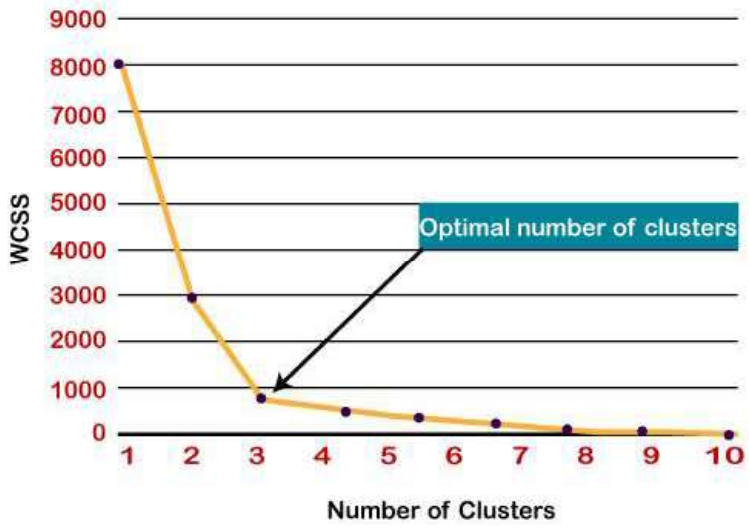
Comments on the K-Means Method

- Strength:
  - Scalable, Efficient: $O(tkn)$, t: # $iterations$, k: # clusters, n: # objects
    - Normally, $k, t << n$.
    - Comparing: PAM: $O(k(n-k)^2)$, $CLARA$: $O(ks2 + k(n-k))$
- Weakness
  - Often terminates at a local optimal.
  - Sensitive to the initial random selection of centroids
  - Applicable only to objects in a continuous n-dimensional space
    - Using the k-modes method for categorical data
    - In comparison, k-medoids can be applied to a wide range of data
  - Need to specify k, the number of clusters, in advance
  - Sensitive to noisy data and outliers
  - Not suitable to discover clusters with non-convex shapes and different densities

Choosing the Appropriate Number of Clusters

$$E = \Sigma_{i=1}^{k} \Sigma_{p \in C_i} (p - c_i)^2$$

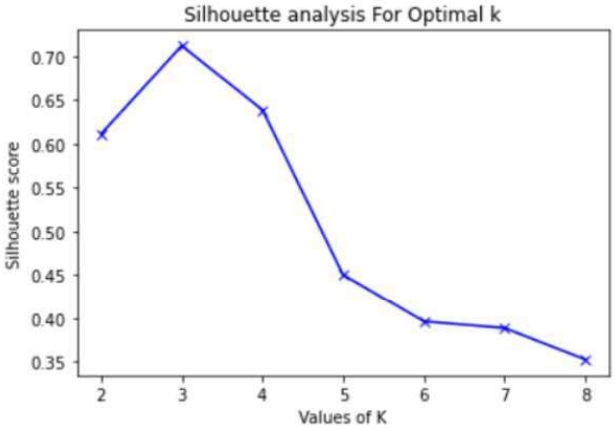
- elbow method
  - uses the concept of WCSS value. **Within Cluster Sum of Squares**, which defines the total variations within a cluster.
  - It executes the K-means clustering on a given dataset for different K values and calculates WCSS
  - Plots a curve between calculated WCSS values and the number of clusters K.
  - The sharp point of bend or a point of the plot looks like an arm, then that point is considered as the best value of K.



Choosing the Appropriate Number of Clusters

$$S(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}}$$

- silhouette coefficient
  - measure of how similar a data point is within-cluster (cohesion) compared to other clusters (separation).
    - a(i) is the average distance between i and all the other data points in the cluster to which i belongs.
    - b(i) is the average distance from i to all clusters to which i does not belong.
  - The value of the silhouette coefficient is between [-1, 1].
  - A score of 1 denotes the best, meaning that the data point i is very compact within the cluster to which it belongs and far away from the other clusters.
  - The worst value is -1. Values near 0 denote overlapping clusters.

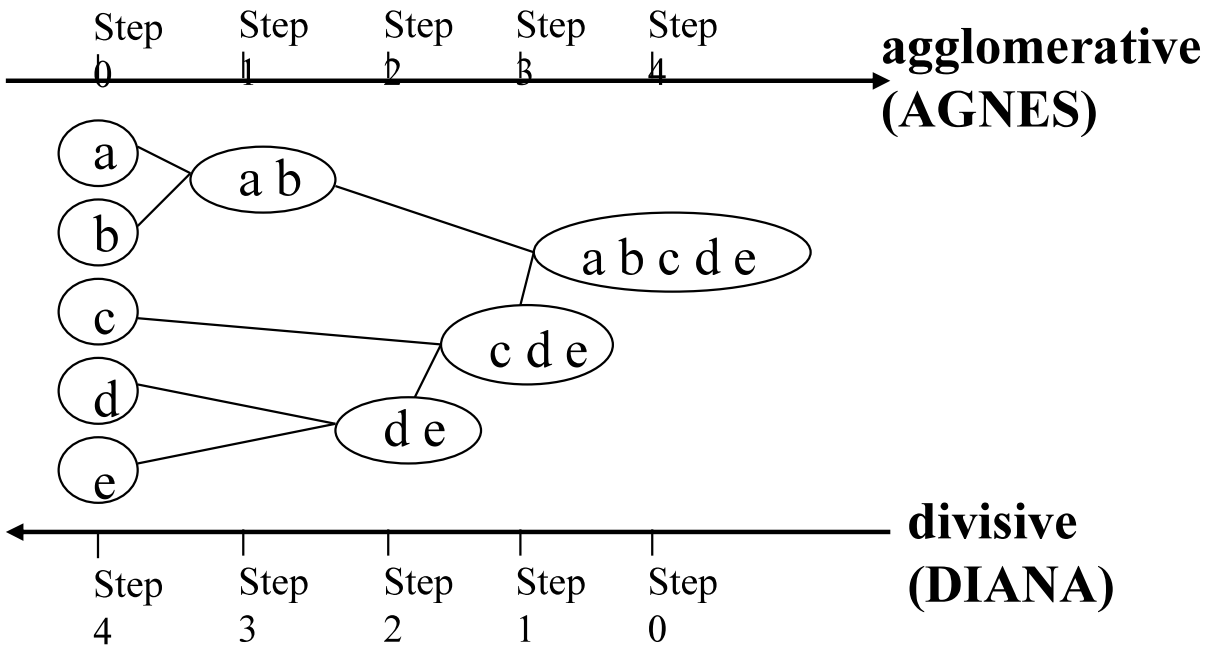


Line plot between K and Silhouette score

Hierarchical Clustering

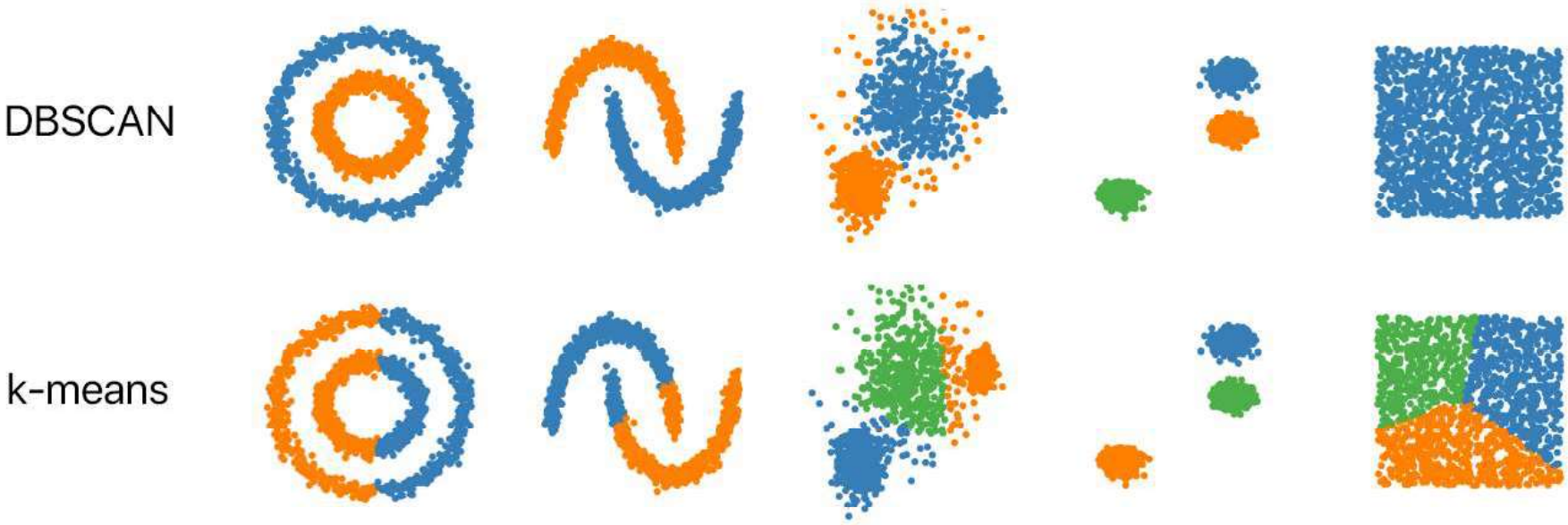
- Produce a set of nested clusters organized as a hierarchical tree. Can be visualized as a dendrogram.
- AGNES (Agglomerative Nesting): bottom-up approach
- DIANA (Divisive Analysis): top-down approach
- hierarchical clustering is a **deterministic** process, meaning cluster assignments won't change when you run an algorithm twice on the same input data.
- Don't have to assume any particular k clusters, any desirable number of clusters can be obtained by cutting the dendrogram.
- Need termination condition
- Use distance matrix as clustering criteria

Hierarchical Clustering



Density-Based Clustering Methods

- Clustering based on density (local cluster criterion), such as density-connected points



Density-Based Clustering Methods

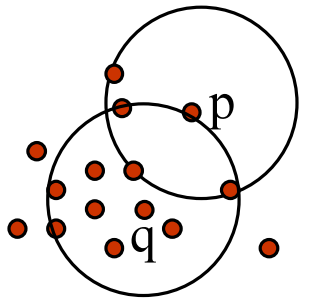
- Clustering based on density (local cluster criterion), such as density-connected points
- Major features:
  - Discover clusters of arbitrary shape
  - doesn't require the user to specify the number of clusters
  - Handle noise
  - One scan
  - Need density parameters as termination condition
- Several interesting studies:
  - DBSCAN: Ester, et al. (KDD'96)
  - OPTICS: Ankerst, et al (SIGMOD'99).
  - DENCLUE: Hinneburg & D. Keim (KDD'98)
  - CLIQUE: Agrawal, et al. (SIGMOD'98) (more grid-based)

DBSCAN

- Finds core point, that is the objects that have dense neighborhoods.
- Connects core point and their neighborhoods to form dense regions as clusters.
- DBSCAN requires only two parameters: ***epsilon*** and ***minPoints***.
- ***Epsilon (Eps)*** is the radius of the circle to be created around each data point to check the density, and ***minPoints (MinPts)*** is the minimum number of data points required inside that circle for that data point to be classified as a Core point.

DBSCAN

- Two parameters:
  - *Eps*: Maximum radius of the neighborhood
  - *MinPts*: Minimum number of points in an Eps-neighbourhood
- *NEps(p): {q belongs to D | dist(p,q) ≤ Eps}*
- Directly density-reachable: A point p is directly density-reachable from a point q w.r.t. Eps, *MinPts* if
  - p belongs to *NEps(q)*
  - q is core point:
  - $|NEps(q)| \geq MinPts$

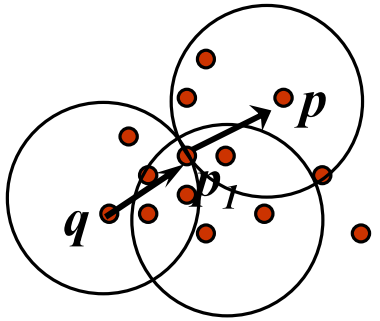


MinPts = 5

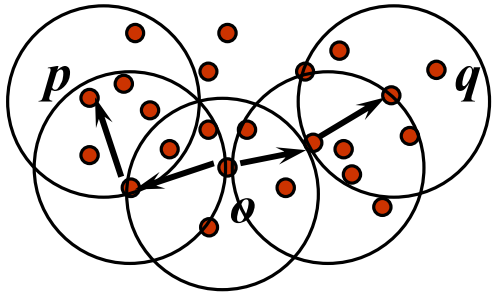
Eps = 1 cm

Density-Reachable and Density-Connected

- Density-reachable:
  - A point p is density-reachable from a point q w.r.t. *Eps, MinPts* if there is a chain of points p_1, …, p_n such that p_1 = q, p_n = p such that p_{i+1} is directly density-reachable from p_i



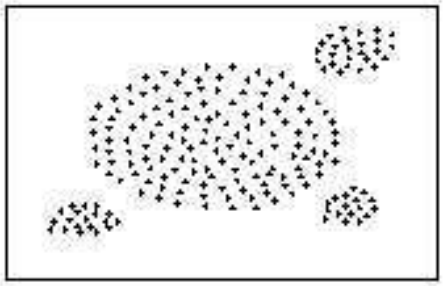
- Density-connected
  - A point p is density-connected to a point q w.r.t. *Eps, MinPts* if there is a point o such that both, p and q are density-reachable from o w.r.t. *Eps* and *MinPts*



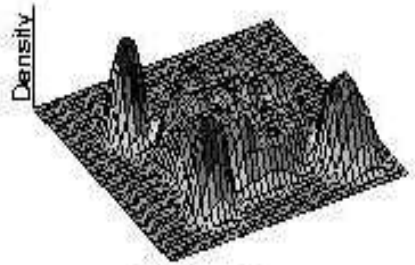
DBSCAN: The Algorithm

- Arbitrary select a point p
- Retrieve all points density-reachable from p w.r.t. *Eps* and *MinPts*
- If p is 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

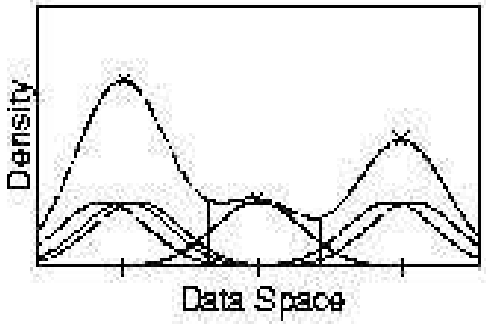
Density Attractor



(a) Data Set



(c) Gaussian



Density-Based Clustering Methods

- **strengths**
  - excel at identifying clusters of nonspherical shapes.
  - resistant to outliers.
- **weaknesses**
  - aren’t well suited for clustering in high-dimensional spaces.
  - have trouble identifying clusters of varying densities.