


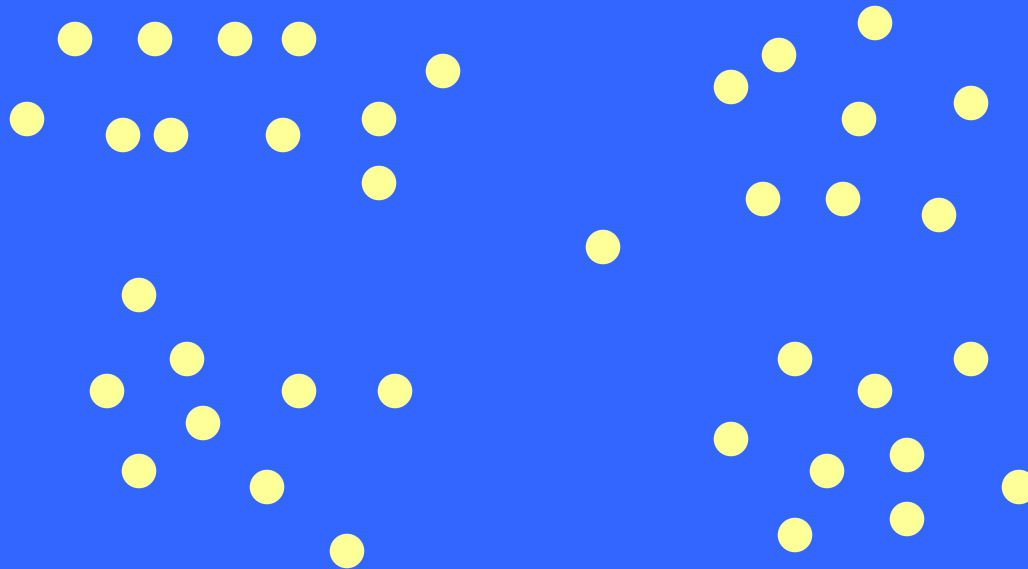
Chapter 5. Cluster Analysis

1. What is Cluster Analysis? 
2. Types of Data in Cluster Analysis
3. A Categorization of Major Clustering Methods
4. Partitioning Methods
5. Hierarchical Methods

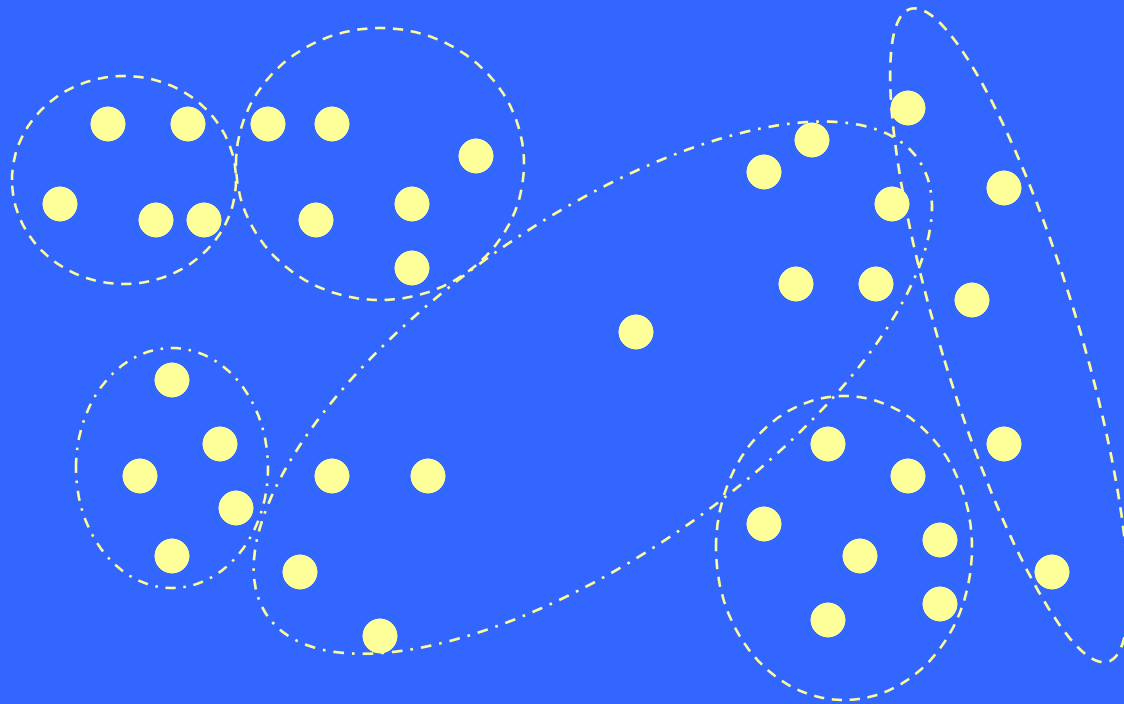
What is Cluster Analysis?

- Cluster: a collection of data objects
 - Similar to one another within the same cluster
 - Dissimilar to the objects in other clusters
- Cluster analysis
 - Finding similarities between data according to the characteristics found in the data and grouping similar data objects into clusters
- **Unsupervised learning**: no predefined classes
- Typical applications
 - As a **stand-alone tool** to get insight into data distribution
 - As a **preprocessing step** for other algorithms

Clustering Houses

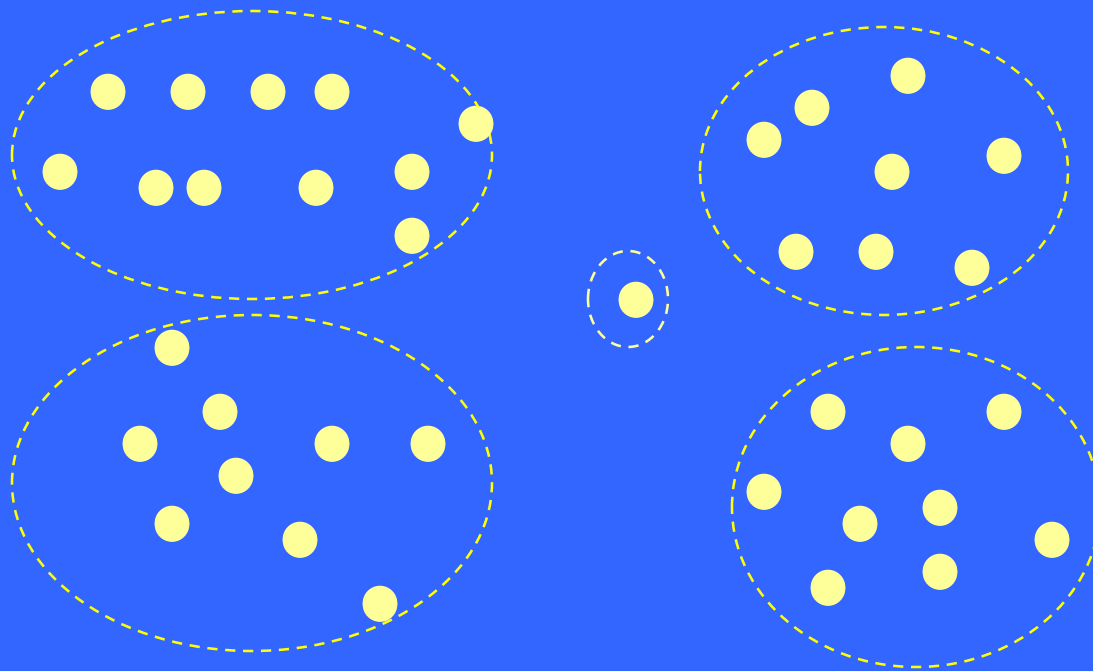


Clustering Houses



Size Based

Clustering Houses



Geographic Distance Based

Clustering: Rich Applications and Multidisciplinary Efforts

- Pattern Recognition
 - Clustering methods simply try to group similar patterns into clusters whose members are more similar to each other
- Spatial Data Analysis
 - Create thematic maps in GIS by clustering feature spaces
 - Detect spatial clusters or for other spatial mining tasks
- Image Processing
 - clustering is used in image segmentation for separating image objects which are analyzed further
- Economic Science (especially market research)
 - cluster analysis is to classify objects into relatively homogeneous groups based on a set of variables considered like demographics, psychographics, buying behaviours, attitudes, preferences, etc.
- WWW
 - Document classification
 - Cluster Weblog data to discover groups of similar access patterns

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
- Earth-quake studies: Observed earth quake epicenters should be clustered along continent faults

Quality: 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

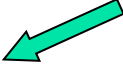
Measure the Quality of Clustering

- **Dissimilarity/Similarity metric**: Similarity is expressed in terms of a distance function, typically metric: $d(i, j)$
- There is a separate “quality” function that measures the “goodness” of a cluster.
- The definitions of **distance functions** are usually very different for interval-scaled, boolean, categorical, ordinal ratio, and vector variables.
- Weights should be associated with different variables based on applications and data semantics.
- It is hard to define “similar enough” or “good enough”
 - the answer is typically highly subjective.

Requirements of Clustering in Data Mining

- Scalability
- Ability to deal with different types of attributes
- Ability to handle dynamic data
- Discovery of clusters with arbitrary shape
- Minimal requirements for domain knowledge to determine input parameters
- Able to deal with noise and outliers
- Insensitive to order of input records
- High dimensionality
- Incorporation of user-specified constraints
- Interpretability and usability

Chapter 5. Cluster Analysis

1. What is Cluster Analysis?
2. Types of Data in Cluster Analysis 
3. A Categorization of Major Clustering Methods
4. Partitioning Methods
5. Hierarchical Methods

Data Structures

- Data matrix
 - (two modes)

$$\begin{bmatrix} x_{11} & \dots & x_{1f} & \dots & x_{1p} \\ \dots & \dots & \dots & \dots & \dots \\ x_{i1} & \dots & x_{if} & \dots & x_{ip} \\ \dots & \dots & \dots & \dots & \dots \\ x_{n1} & \dots & x_{nf} & \dots & x_{np} \end{bmatrix}$$

- Dissimilarity matrix
 - (one mode)

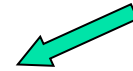
$$\begin{bmatrix} 0 & & & & \\ d(2,1) & 0 & & & \\ d(3,1) & d(3,2) & 0 & & \\ \vdots & \vdots & \vdots & & \\ d(n,1) & d(n,2) & \dots & \dots & 0 \end{bmatrix}$$

Type of data in clustering analysis

- Interval-scaled variables
- Binary variables
- Nominal, ordinal, and ratio variables
- Variables of mixed types

Chapter 5. Cluster Analysis

1. What is Cluster Analysis?
2. Types of Data in Cluster Analysis
3. A Categorization of Major Clustering Methods
4. Partitioning Methods
5. Hierarchical Methods



Major Clustering Approaches (I)

- 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, ROCK, CAMELEON
- Density-based approach:
 - Based on connectivity and density functions
 - Typical methods: DBSCAN, OPTICS, DenClue

Major Clustering Approaches (II)

- Grid-based approach:
 - based on a multiple-level granularity structure
 - Typical methods: STING, WaveCluster, CLIQUE
- 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: pCluster
- User-guided or constraint-based:
 - Clustering by considering user-specified or application-specific constraints
 - Typical methods: COD (obstacles), constrained clustering

Typical Alternatives to Calculate the Distance between Clusters

- Single link: smallest distance between an element in one cluster and an element in the other, i.e., $\text{dis}(K_i, K_j) = \min(t_{ip}, t_{jq})$
- Complete link: largest distance between an element in one cluster and an element in the other, i.e., $\text{dis}(K_i, K_j) = \max(t_{ip}, t_{jq})$
- Average: avg distance between an element in one cluster and an element in the other, i.e., $\text{dis}(K_i, K_j) = \text{avg}(t_{ip}, t_{jq})$
- Centroid: distance between the centroids of two clusters, i.e., $\text{dis}(K_i, K_j) = \text{dis}(C_i, C_j)$
- Medoid: distance between the medoids of two clusters, i.e., $\text{dis}(K_i, K_j) = \text{dis}(M_i, M_j)$
 - Medoid: one chosen, centrally located object in the cluster

Centroid, Radius and Diameter of a Cluster (for numerical data sets)

- Centroid: the “middle” of a cluster

$$C_m = \frac{\sum_{i=1}^N (t_{ip})}{N}$$


- Radius: square root of average distance from any point of the cluster to its centroid

$$R_m = \sqrt{\frac{\sum_{i=1}^N (t_{ip} - c_m)^2}{N}}$$

- Diameter: square root of average mean squared distance between all pairs of points in the cluster

$$D_m = \sqrt{\frac{\sum_{i=1}^N \sum_{q=1}^N (t_{ip} - t_{iq})^2}{N(N-1)}}$$

Chapter 5. Cluster Analysis

1. What is Cluster Analysis?
2. Types of Data in Cluster Analysis
3. A Categorization of Major Clustering Methods
4. Partitioning Methods 
5. Hierarchical Methods

Partitioning Algorithms: Basic Concept

- Partitioning method: Construct a partition of a database ***D*** of ***n*** objects into a set of ***k*** clusters, s.t., min sum of squared distance

$$\sum_{m=1}^k \sum_{t_{mi} \in K_m} (C_m - t_{mi})^2$$

- Given a *k*, find a partition of *k clusters* that optimizes the chosen partitioning criterion
 - Global optimal: exhaustively enumerate all partitions
 - Heuristic methods: *k-means* and *k-medoids* algorithms
 - *k-means* (MacQueen'67): Each cluster is represented by the center of the cluster
 - *k-medoids* or PAM (Partition around medoids) (Kaufman & Rousseeuw'87): Each cluster is represented by one of the objects in the cluster

The *K-Means* Clustering Method

- Given k , the *k-means* algorithm is implemented in four steps:
 - Partition objects into k nonempty subsets
 - Compute seed points as the centroids of the clusters of the current partition (the centroid is the center, i.e., *mean point*, of the cluster)
 - Assign each object to the cluster with the nearest seed point
 - Go back to Step 2, stop when no more new assignment

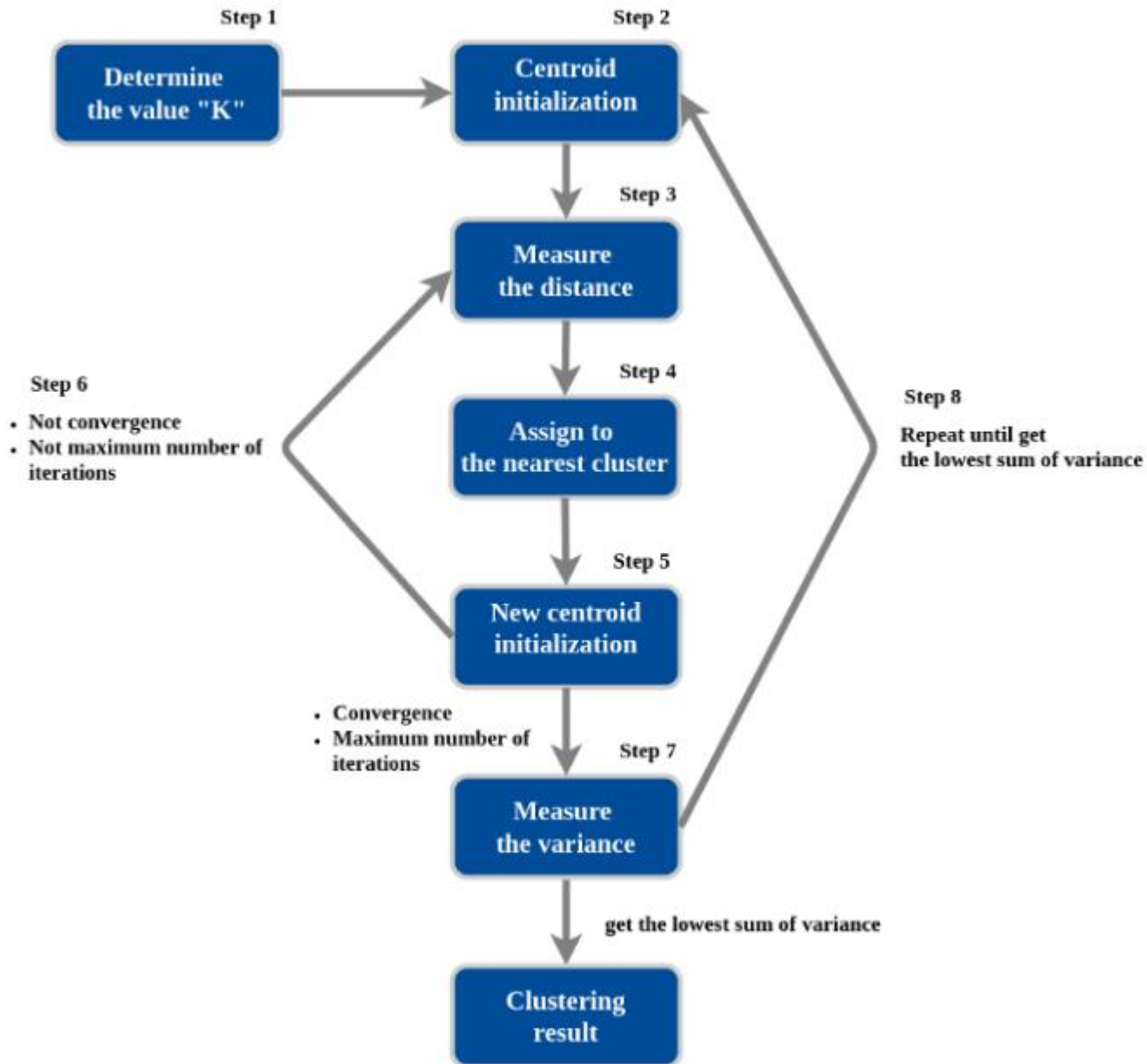
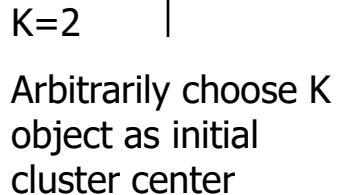


Image Source: <https://medium.com/data-folks-indonesia/step-by-step-to-understanding-k-means-clustering-and-implementation-with-sklearn-b55803f519d6>

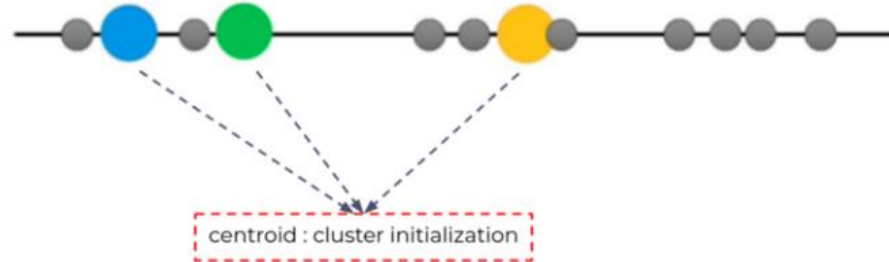
- Example



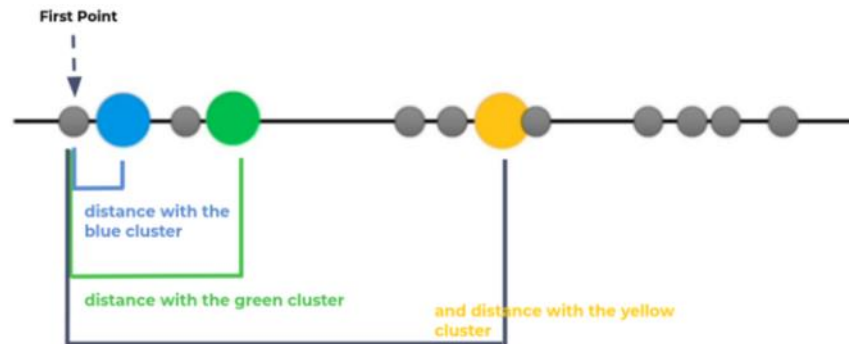
The *K-Means* Clustering Method

Step1: Select value of k

Step2: Initialize cluster



Step3: Measure the Euclidean distance between each point and centroid



Step4: Assign each point to the nearest cluster



Do the same treatment for the other unlabeled point, until we get this

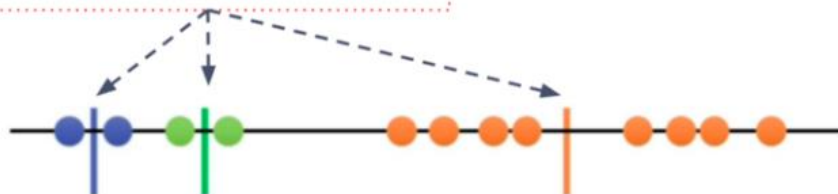


Assign the each point to the nearest cluster

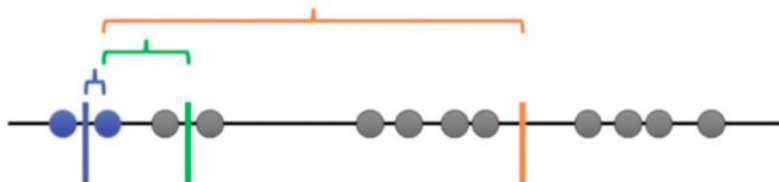


Step5: Calculate the mean of each cluster as new centroid

Calculate the mean of each cluster



Step6: Repeat step 3–5 with the new center of cluster

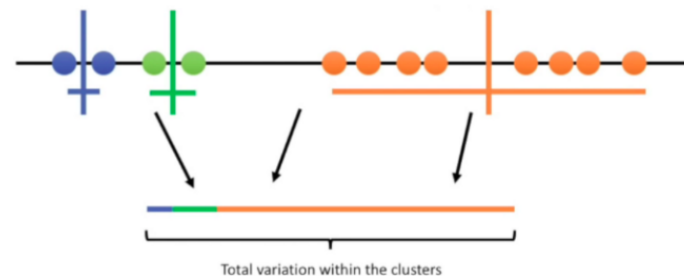


Repeat until stop:

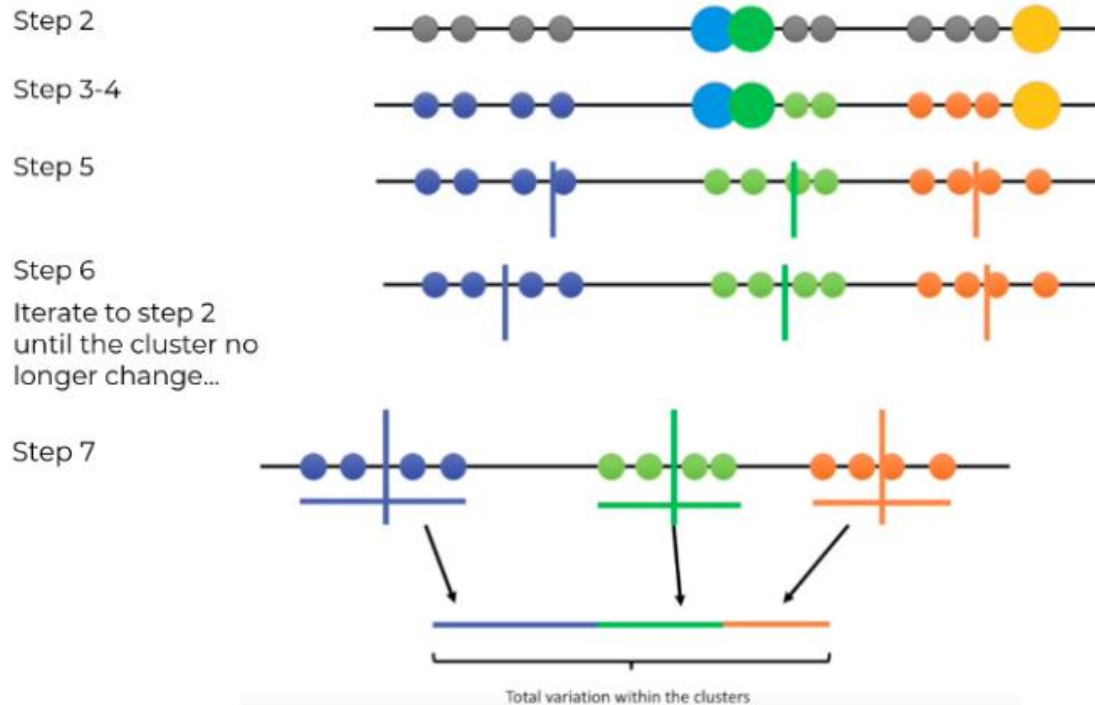
- Convergence. (No further changes)
- Maximum number of iterations.



Step7: Calculate the variance of each cluster



Step8: Repeat step 2–7 until get the lowest sum of variance



For example — attempts 3 with different random centroid



Repeat until stop:

- Until we get the lowest sum of variance and pick those cluster as our result



K-Means example

- Cluster the following items in 2 clusters: {2, 4, 10, 12, 3, 20, 30, 11, 25}
- $d(C_i, t_i) = \sqrt{(C_i - t_i)^2}$
- Assignment to K = $\min (d(C_i, t_i))$

M1	M2	K1	K2
2	4	{2,3}	{4,10, 12, 20, 30, 11, 25}
2.5	16	{2, 3, 4}	{10, 12, 20, 30, 11, 25}
3	18	{2, 3, 4, 10}	{12, 20, 30, 11, 25}
4.75	19.6	{2, 3, 4, 10, 12, 11}	{20, 30, 25}
7	25	{2, 3, 4, 10, 12, 11}	{20, 30, 25}

Stopping Criteria:

- No new assignment
- No change in cluster means

K-means 2D example

- Apply k-means for the following dataset to make 2 clusters:

X	Y
185	72
170	56
168	60
179	68
182	72
188	77

Step 1: Assume Initial Centroids: $C1 = (185, 72)$, $C2 = (170, 56)$

Step 2: Calculate Euclidean Distance to each centroid:

$$d[(x, y), (a, b)] = \sqrt{(x - a)^2 + (y - b)^2}$$

For $t1 = (168, 60)$

$$\begin{aligned} d[(185, 72), (168, 60)] &= \sqrt{(185 - 168)^2 + (72 - 60)^2} \\ &= 20.808 \end{aligned}$$

$$\begin{aligned} d[(170, 56), (168, 60)] &= \sqrt{(170 - 168)^2 + (56 - 60)^2} \\ &= 4.472 \end{aligned}$$

Since $d(C2, t1) < d(C1, t1)$. So assign $t1$ to $C2$

Step 3: For $t2 = (179, 68)$

$$\begin{aligned} d[(185, 72), (179, 68)] &= \sqrt{(185 - 179)^2 + (72 - 68)^2} \\ &= 7.211 \end{aligned}$$

$$\begin{aligned} d[(170, 56), (179, 68)] &= \sqrt{(170 - 179)^2 + (56 - 68)^2} \\ &= 15 \end{aligned}$$

Since $d(C1, t2) < d(C2, t2)$ So assign $t2$ to $C1$

Step 4: For $t3 = (182, 72)$

$$\begin{aligned} d[(185, 72), (182, 72)] &= \sqrt{(185 - 182)^2 + (72 - 72)^2} \\ &= 3 \end{aligned}$$

$$\begin{aligned} d[(170, 56), (182, 72)] &= \sqrt{(170 - 182)^2 + (56 - 72)^2} \\ &= 20 \end{aligned}$$

Since $d(C1, t3) < d(C2, t3)$, So assign $t3$ to $C1$

K-means 2D example

- Apply k-means for the following dataset to make 2 clusters:

X	Y
185	72
170	56
168	60
179	68
182	72
188	77

Step 5: For $t_4 = (188, 77)$

$$d[(185, 72), (182, 72)] = \sqrt{(185 - 188)^2 + (72 - 77)^2} \\ = 5.83$$

$$d[(170, 56), (182, 72)] = \sqrt{(170 - 188)^2 + (56 - 77)^2} \\ = 27.65$$

Since $d(C_1, t_4) < d(C_2, t_4)$, So assign t_4 to C_1

Step 6: Clusters after 1 iteration

$D_1 = \{(185, 72), (179, 68), (182, 72), (188, 77)\}$

$D_2 = \{(170, 56), (168, 60)\}$

Step 7: New clusters centroids $C_1 = \{183.5, 72.25\}$ $C_2 = \{169, 58\}$

Repeat above steps for all samples till convergence

K-means 2D example

- Apply k-means for the following dataset to make 2 clusters:

X	Y
185	72
170	56
168	60
179	68
182	72
188	77

Step 2: Calculate Euclidean Distance to each centroid:

$$d[(x, y), (a, b)] = \sqrt{(x - a)^2 + (y - b)^2}$$

For $t_1 = (168, 60)$

$$d[(183.5, 72.25), (168, 60)] = \sqrt{(183.5 - 168)^2 + (72.25 - 60)^2}$$

$$= \text{xxxxx}$$

$$d[(169, 58), (168, 60)] = \sqrt{(169 - 168)^2 + (58 - 60)^2}$$

$$= \text{xxxxx}$$

Since $d(C_2, t_1) < d(C_1, t_1)$. So assign t_1 to C

Step 3: For $t_2 = (179, 68)$

$$d[(183.5, 72.25), (179, 68)] = \sqrt{(183.5 - 179)^2 + (72.25 - 68)^2}$$

$$= \text{xxxxx}$$

$$d[(169, 58), (179, 68)] = \sqrt{(169 - 179)^2 + (58 - 68)^2}$$

$$= \text{xxxxx}$$

Since $d(C_1, t_2) < d(C_2, t_2)$ So assign t_2 to

Step 4: For $t_3 = (182, 72)$

$$d[(183.5, 72.25), (182, 72)] = \sqrt{(183.5 - 182)^2 + (72.25 - 72)^2}$$

$$= \text{xxxxx}$$

$$d[(169, 58), (182, 72)] = \sqrt{(169 - 182)^2 + (58 - 72)^2}$$

$$= \text{xxxxx}$$

Since $d(C_1, t_3) < d(C_2, t_3)$, So assign t_3 to

K-means 2D example

- Apply k-means for the following dataset to make 2 clusters:

X	Y
185	72
170	56
168	60
179	68
182	72
188	77

Step 6: Clusters after 2 iteration

$D1 = \{(185, 72), (179, 68), (182, 72), (188, 77)\}$

$D2 = \{(170, 56), (168, 60)\}$

Step 7: New clusters centroids $C1 = \{ \quad \quad \quad \}$ $C2 = \{ \quad \quad \quad \}$

Repeat above steps for all samples till convergence

Final Clusters

$D1 = \{(185, 72), (179, 68), (182, 72), (188, 77)\}$

$D2 = \{(170, 56), (168, 60)\}$

Comments on the *K-Means* Method

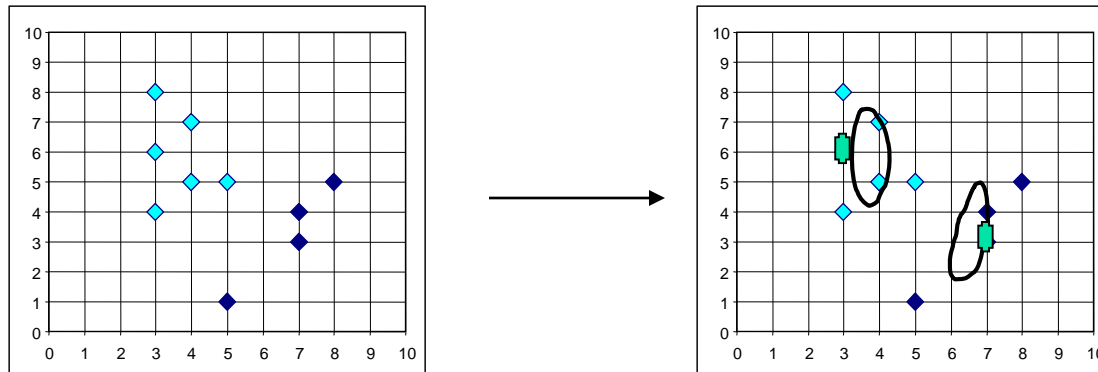
- Strength: *Relatively efficient*: $O(tkn)$, where n is # objects, k is # clusters, and t is # iterations. Normally, $k, t \ll n$.
 - Comparing: PAM: $O(k(n-k)^2)$, CLARA: $O(ks^2 + k(n-k))$
- Comment: Often terminates at a *local optimum*. The *global optimum* may be found using techniques such as: *deterministic annealing* and *genetic algorithms*
- Weakness
 - 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*
 - Not suitable to discover clusters with *non-convex shapes*

Variations of the *K-Means* Method

- A few variants of the *k-means* which differ in
 - Selection of the initial k means
 - Dissimilarity calculations
 - Strategies to calculate cluster means
- Handling categorical data: *k-modes* (Huang'98)
 - Replacing means of clusters with modes
 - Using new dissimilarity measures to deal with categorical objects
 - Using a frequency-based method to update modes of clusters
 - A mixture of categorical and numerical data: *k-prototype* method

What Is the Problem of the K-Means Method?

- The k-means algorithm is sensitive to outliers !
 - Since an object with an extremely large value may substantially distort the distribution of the data.
- K-Medoids: Instead of taking the **mean** value of the object in a cluster as a reference point, **medoids** can be used, which is the **most centrally located** object in a cluster.



The *K-Medoids* Clustering Method

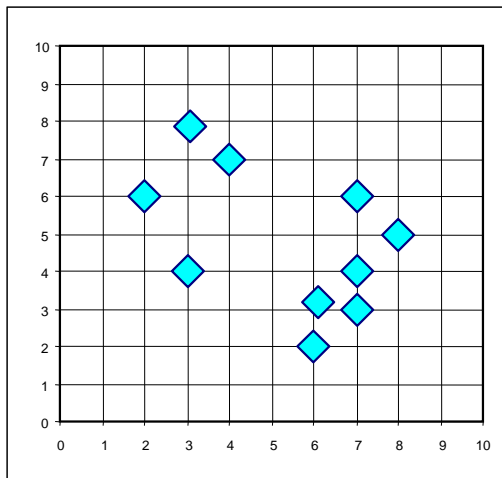
- Find *representative* objects, called medoids, in clusters
- *PAM* (Partitioning Around Medoids, 1987)
 - starts from an initial set of medoids and iteratively replaces one of the medoids by one of the non-medoids if it improves the total distance of the resulting clustering
 - *PAM* works effectively for small data sets, but does not scale well for large data sets
- *CLARA* (Kaufmann & Rousseeuw, 1990) – Clustering LARge Applications
- *CLARANS* (Ng & Han, 1994): Clustering Large Applications based upon RANdomized Search
- Focusing + spatial data structure (Ester et al., 1995)

PAM (Partitioning Around Medoids) (1987)

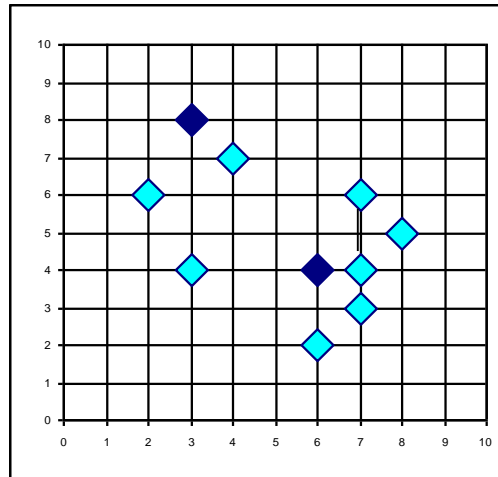
- PAM (Kaufman and Rousseeuw, 1987), built in Splus
- Use real object to represent the cluster
 1. Select k representative objects arbitrarily
 2. For each pair of non-selected object h and selected object i , calculate the total swapping cost TC_{ih}
 3. For each pair of i and h ,
 - If $TC_{ih} < 0$, i is replaced by h
 - Then assign each non-selected object to the most similar representative object
 4. repeat steps 2-3 until there is no change

A Typical K-Medoids Algorithm (PAM)

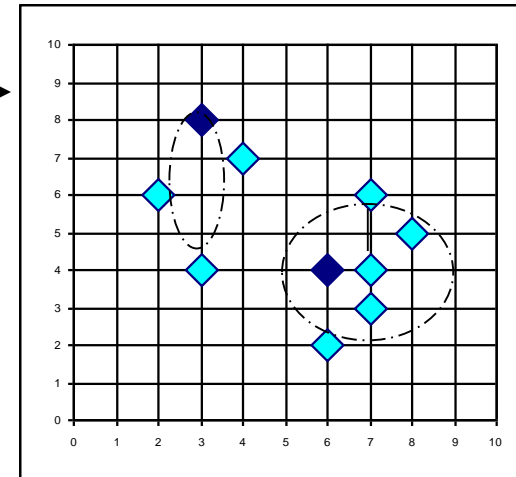
Total Cost = 20



Arbitrary
choose k
object as
initial
medoids



Assign
each remainin
g object to
nearest
medoids

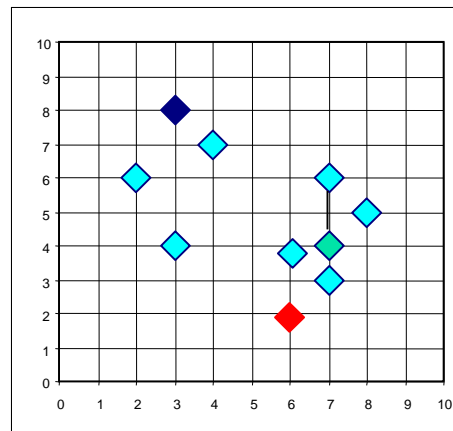


$K=2$

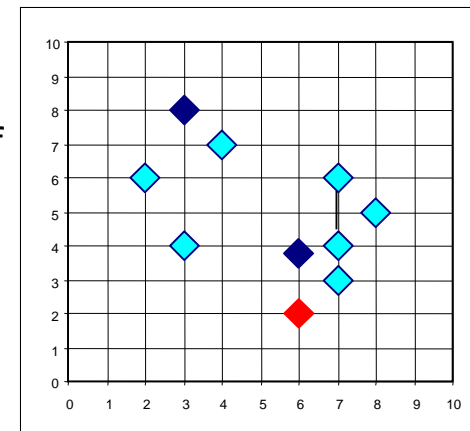
Do loop
Until no
change

Swapping O
and O_{random}
If quality is
improved.

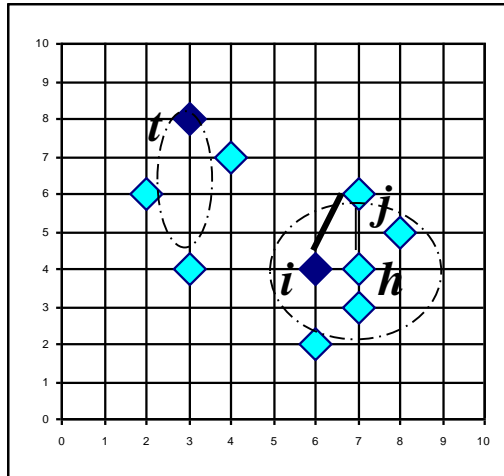
Total Cost = 26



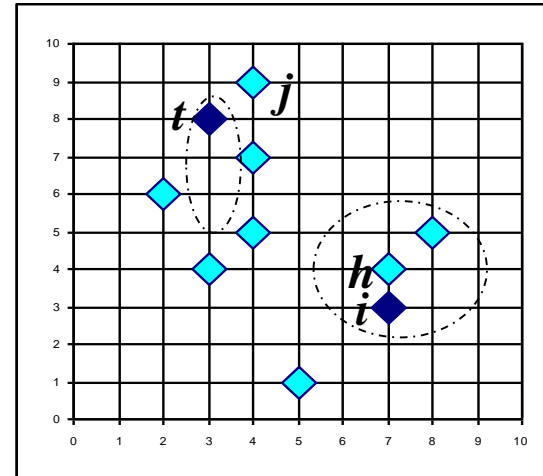
Compute
total cost of
swapping



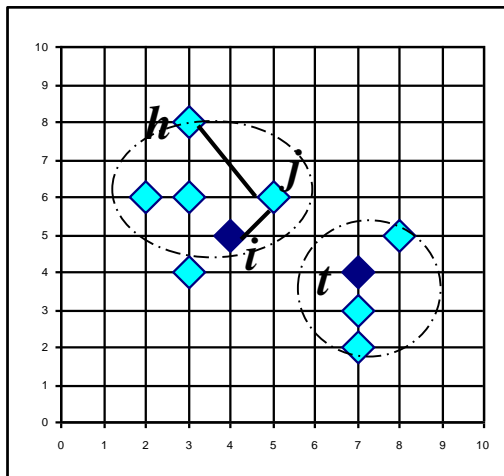
PAM Clustering: Total swapping cost $TC_{ih} = \sum_j C_{jih}$



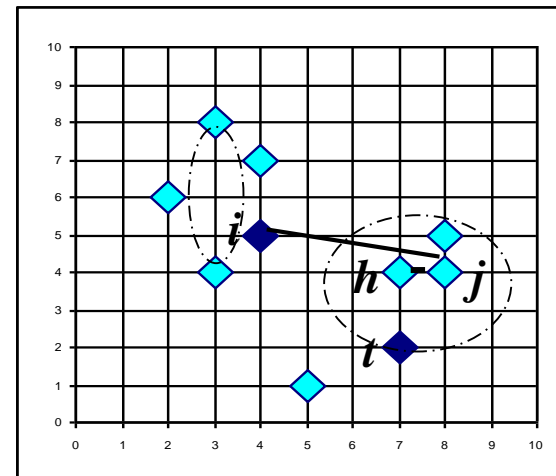
$$C_{jih} = d(j, h) - d(j, i)$$



$$C_{jih} = 0$$



$$C_{jih} = d(j, t) - d(j, i)$$



$$C_{jih} = d(j, h) - d(j, i)$$

PAM example

	A	B	C	D	E
A	0	1	2	2	3
B	1	0	2	4	3
C	2	2	0	1	5
D	2	4	1	0	3
E	3	3	5	3	0

Step 1: Let A & B be mediod. Obtain clusters by assigning elements close to the respective mediods

Step 1: {A, C, D} & {B, E}

Step 2: Now examine the three non-mediod {C, D, E} to determine if they can replace existing mediods.

i.e. A replaced by C, D, or E and B replaced by C, D or E

We have 6 costs to determine

TC_{AC} , TC_{AD} , TC_{AE} , TC_{BC} , TC_{BD} , TC_{BE}

Let us replace A with C. Cost of replacing A with C

$TC_{AC} = C_{AAC} + C_{BAC} + C_{CAC} + C_{DAC} + C_{EAC}$

C_{jih} = cost change for item t_j associated by swapping mediod t_i with non-mediod t_j

{A, B, E} & {C, D}

$TC_{AC} = 1 + 0 + (-2) + (-1) + 0 = -2$

The overall cluster is reduced by 2

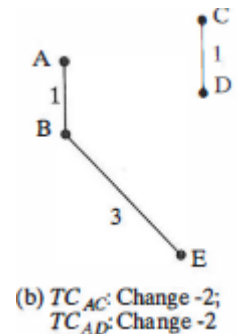
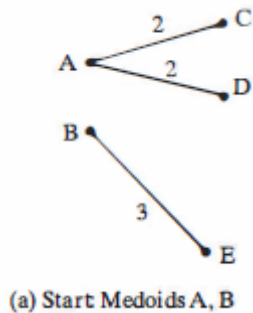
Step 3: Let us replace A with D. Cost of replacing A with D

{A, B, E} & {C, D}

$TC_{AD} = C_{AAD} + C_{BAD} + C_{CAD} + C_{DAD} + C_{EAD}$

$TC_{AD} = 1 + 0 + (-1) + (-2) + 0 = -2$

The overall cluster is reduced by 2



	A	B	C	D	E
A	0	1	2	2	3
B	1	0	2	4	3
C	2	2	0	1	5
D	2	4	1	0	3
E	3	3	5	3	0

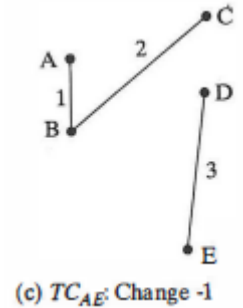
Step 4: Let us replace A with E. Cost of replacing A with E

{A, B, C} & {E, D}

$$TC_{AE} = C_{AAE} + C_{BAE} + C_{CAE} + C_{DAE} + C_{EAE}$$

$$TC_{AD} = 1 + 0 + 0 + 1 + (-3) = -1$$

The overall cluster is reduced by 1



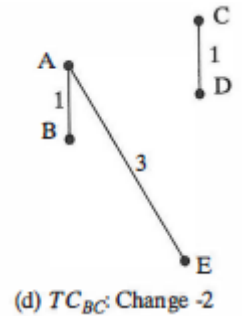
Step 5: Let us replace B with C. Cost of replacing B with C

{A, B, E} & {C, D}

$$TC_{BC} = C_{ABC} + C_{BBC} + C_{CBC} + C_{DBC} + C_{EBC}$$

$$TC_{BC} = 0 + 1 + (-2) + (-1) + 0 = -2$$

The overall cluster is reduced by 2



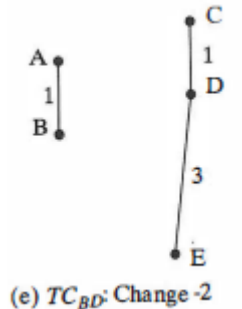
Step 6: Let us replace B with D. Cost of replacing B with D

{A, B} & {C, D, E}

$$TC_{BD} = C_{ABD} + C_{BBD} + C_{CBD} + C_{DBD} + C_{EBD}$$

$$TC_{BD} = 0 + 1 + (-1) + (-2) + 0 = -2$$

The overall cluster is reduced by 2



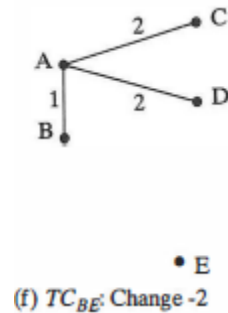
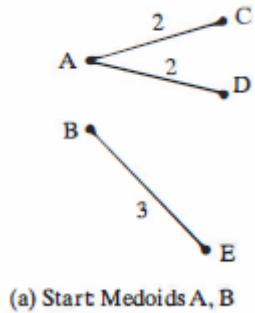
Step 7: Let us replace B with E. Cost of replacing B with E

{A, B, C, D} & {E}

$$TC_{BE} = C_{ABE} + C_{BBE} + C_{CBE} + C_{DBE} + C_{EBE}$$

$$TC_{BE} = 0 + 1 + 0 + 0 + (-3) = -2$$

The overall cluster is reduced by 2



What Is the Problem with PAM?

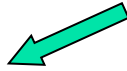
- Pam is more robust than k-means in the presence of noise and outliers because a medoid is less influenced by outliers or other extreme values than a mean
- Pam works efficiently for small data sets but does not **scale well** for large data sets.
 - $O(k(n-k)^2)$ for each iteration

where n is # of data, k is # of clusters

➔ Sampling based method,
CLARA(Clustering LARge Applications)

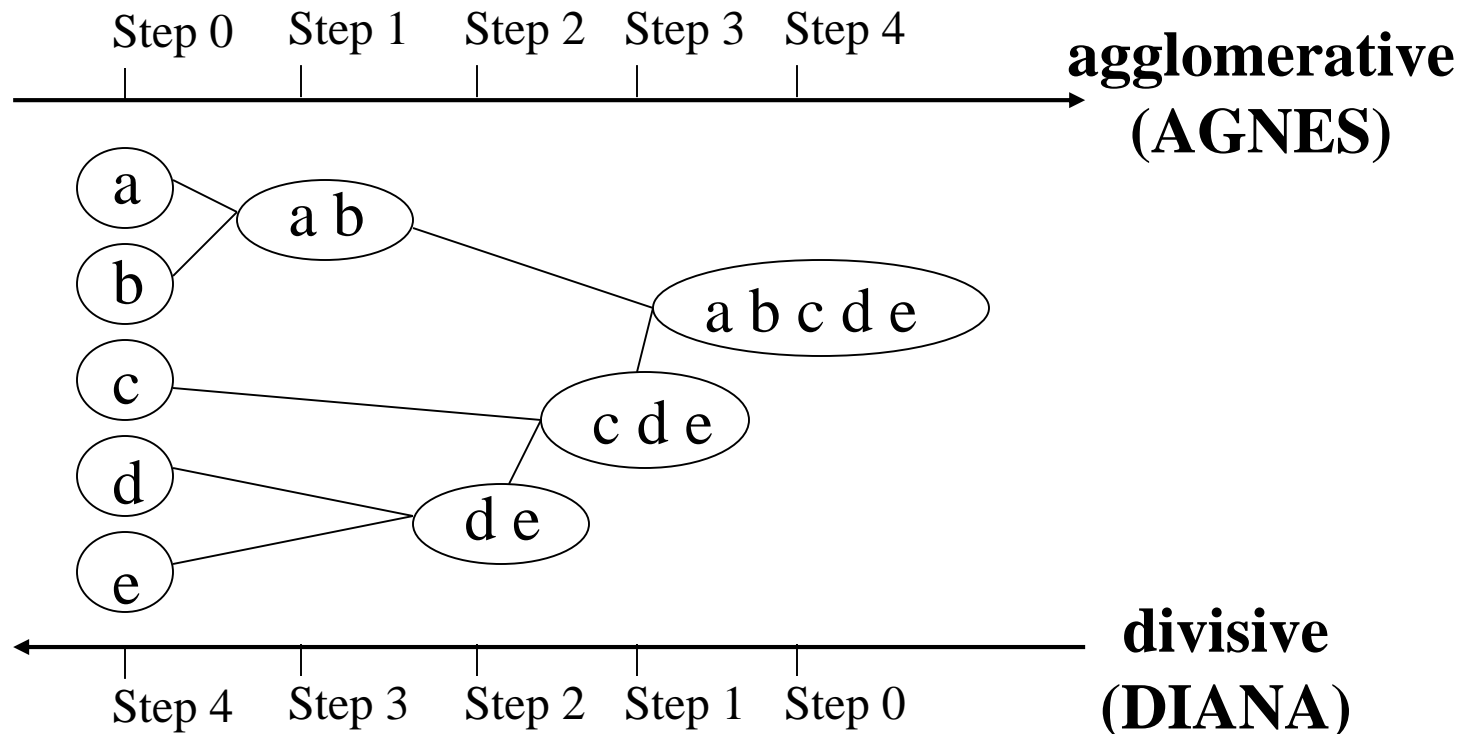
Chapter 5. Cluster Analysis

1. What is Cluster Analysis?
2. Types of Data in Cluster Analysis
3. A Categorization of Major Clustering Methods
4. Partitioning Methods
5. Hierarchical Methods



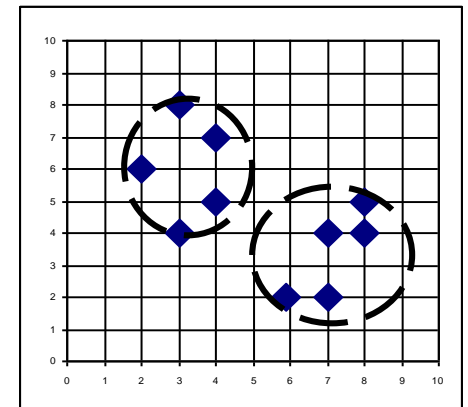
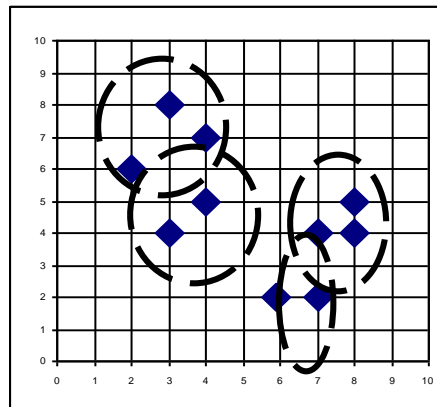
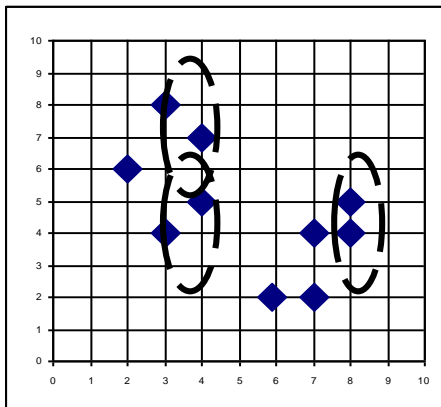
Hierarchical Clustering

- Use distance matrix as clustering criteria. This method does not require the number of clusters k as an input, but needs a termination condition

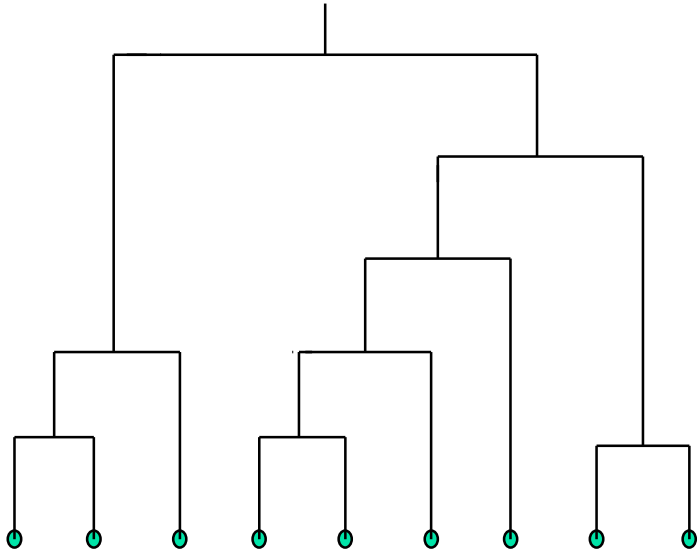


AGNES (Agglomerative Nesting)

- Introduced in Kaufmann and Rousseeuw (1990)
- Implemented in statistical analysis packages, e.g., Splus
- Use the Single-Link method and the dissimilarity matrix.
- Merge nodes that have the least dissimilarity
- Go on in a non-descending fashion
- Eventually all nodes belong to the same cluster



Dendrogram: Shows How the Clusters are Merged



Decompose data objects into a several levels of nested partitioning (tree of clusters), called a dendrogram.

A clustering of the data objects is obtained by cutting the dendrogram at the desired level, then each connected component forms a cluster.

Hierarchical Clustering Single Link example

	A	B	C	D	E
A	0	1	2	2	3
B	1	0	2	4	3
C	2	2	0	1	5
D	2	4	1	0	3
E	3	3	5	3	0

Step 1: At level 0, 5 clusters

Step 1: {A}, {B}, {C}, {D}, {E}

Step 2: At Level 1: Min_dist = 1,

Find distance between each pair. If the distance between elements is \leq min_dist then merge them into one cluster

If $\min_dist\{t_i, t_j\} \leq 1$, then merge clusters

{A, B}, {C, D}, {E}

Step 3: At Level 2: Min_dist = 2,

Find distance between clusters formed in step 2

A \rightarrow C = 2 B \rightarrow C = 2

A \rightarrow D = 2 B \rightarrow D = 4

Hence $\min_dist(\{A, B\}, \{C, D\}) = 2$

A \rightarrow E = 3 B \rightarrow E = 3 $\min_dist(\{A, B\}, \{E\}) = 3$

C \rightarrow E = 5 D \rightarrow E = 3 $\min_dist(\{C, D\}, \{E\}) = 3$

Since threshold is 2, we merge {A, B, C, D}, {E}

Hierarchical Clustering Single Link example

	A	B	C	D	E
A	0	1	2	2	3
B	1	0	2	4	3
C	2	2	0	1	5
D	2	4	1	0	3
E	3	3	5	3	0

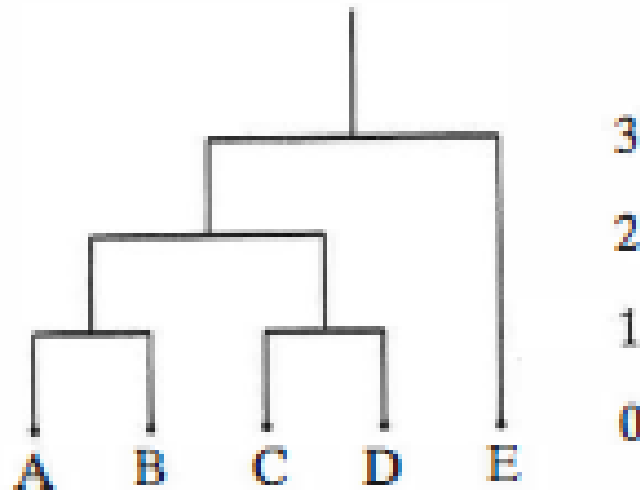
Step 4: At Level 3: $\text{Min_dist} = 3$,

Find distance between clusters formed in step 3

$A \rightarrow E = 3$ $B \rightarrow E = 3$ $C \rightarrow E = 5$ $D \rightarrow E = 3$

$\text{min_dist}(\{A, B, C, D\}, \{E\}) = 3$

Since threshold is 3, we merge both the clusters to get $\{A, B, C, D, E\}$



(a) Single link

Hierarchical Clustering Complete Link example

	A	B	C	D	E
A	0	1	2	2	3
B	1	0	2	4	3
C	2	2	0	1	5
D	2	4	1	0	3
E	3	3	5	3	0

Step 1: At level 0, 5 clusters

Step 1: {A}, {B}, {C}, {D}, {E}

Step 2: At Level 1: Max_dist = 1,

Find distance between each pair.

If $\max_dist\{t_i, t_j\} \leq 1$, then merge clusters

{A, B}, {C, D}, {E}

Step 3: At Level 2: Max_dist = 2,

Find distance between clusters formed in step 2

A->C = 2 B->C = 2

A->D = 2 B->D = 4

Hence $\max_dist(\{A, B\}, \{C, D\}) = 4$

A->E = 3 B->E = 3 $\max_dist(\{A, B\}, \{E\}) = 3$

C->E = 5 D->E = 3 $\max_dist(\{C, D\}, \{E\}) = 5$

Since threshold is 2, no merge at this level

Hierarchical Clustering Complete Link example

	A	B	C	D	E
A	0	1	2	2	3
B	1	0	2	4	3
C	2	2	0	1	5
D	2	4	1	0	3
E	3	3	5	3	0

Step 4: At Level 3: Max_dist = 3,

Find distance between clusters formed in step 2

A->C = 2 B->C = 2

A->D = 2 B->D = 4

Hence $\max_dist(\{A,B\}, \{C, D\}) = 4$

A->E = 3 B->E = 3 $\max_dist(\{A,B\}, \{E\}) = 3$

C->E = 5 D->E = 3 $\max_dist(\{C, D\}, \{E\}) = 5$

Since threshold is 3, $\max_dist(\{A,B\}, \{E\})$ is 3 so we merge them

Step 5: At Level 4: Max_dist = 4,

Find distance between clusters formed in step 3

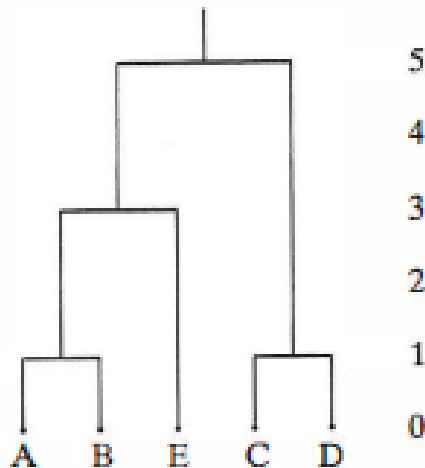
A->C = 2 B->C = 2 A->D = 2 B->D = 4

C->E = 5 D->E = 3

$\max_dist(\{C, D\}, \{A, B, E\}) = 5$

Since threshold is 4, So no merge

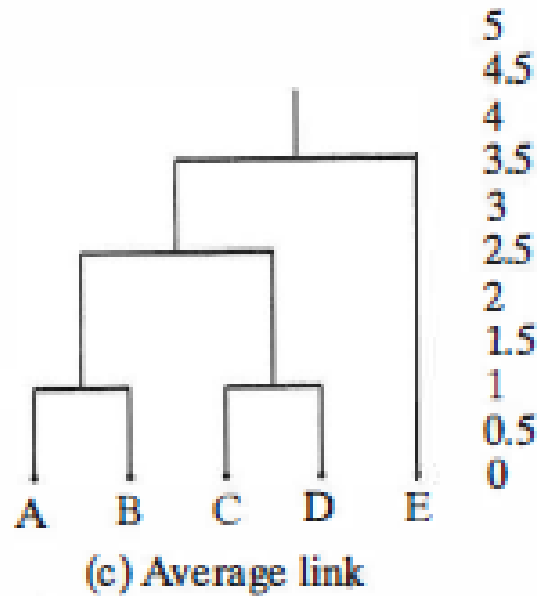
Step 6: At Level 5: Merge both clusters.



(b) Complete link

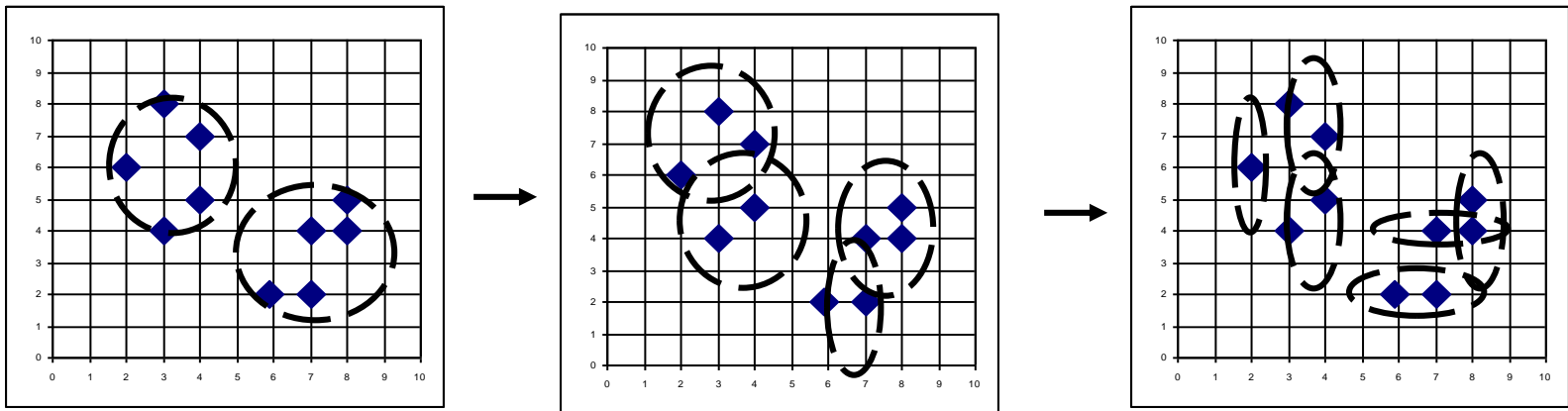
Hierarchical Clustering Average Link example

	A	B	C	D	E
A	0	1	2	2	3
B	1	0	2	4	3
C	2	2	0	1	5
D	2	4	1	0	3
E	3	3	5	3	0



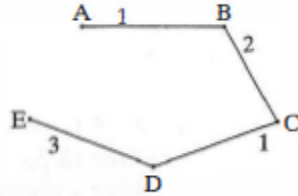
DIANA (Divisive Analysis)

- Introduced in Kaufmann and Rousseeuw (1990)
- Implemented in statistical analysis packages, e.g., Splus
- Inverse order of AGNES
- Eventually each node forms a cluster on its own



Hierarchical Clustering Divisive example

	A	B	C	D	E
A	0	1	2	2	3
B	1	0	2	4	3
C	2	2	0	1	5
D	2	4	1	0	3
E	3	3	5	3	0



Step 1: Largest edge is with dist 3. Cutting at largest edge

Step 1: {A, B, C, D} and {E}

Step 2: Now split the edge BC,

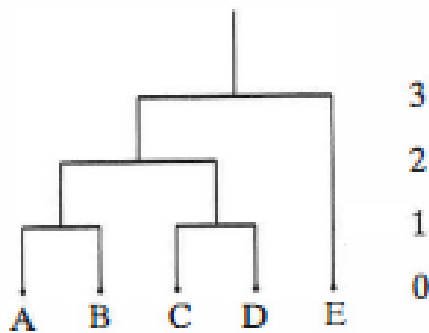
{A, B}, {C, D}, {E}

Step 3: Now split the edge BC,

{A, B}, {C, D}, {E}

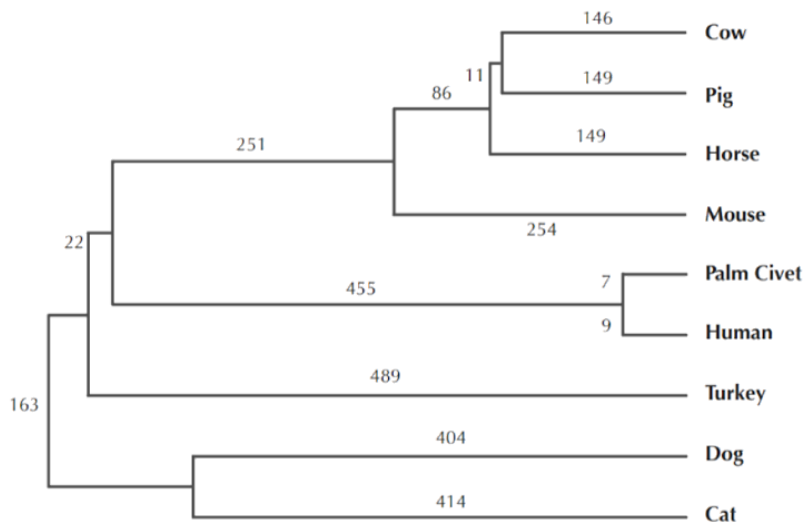
Step 4: Remaining edges are split to create clusters ,

{A}, {B}, {C}, {D}, {E}



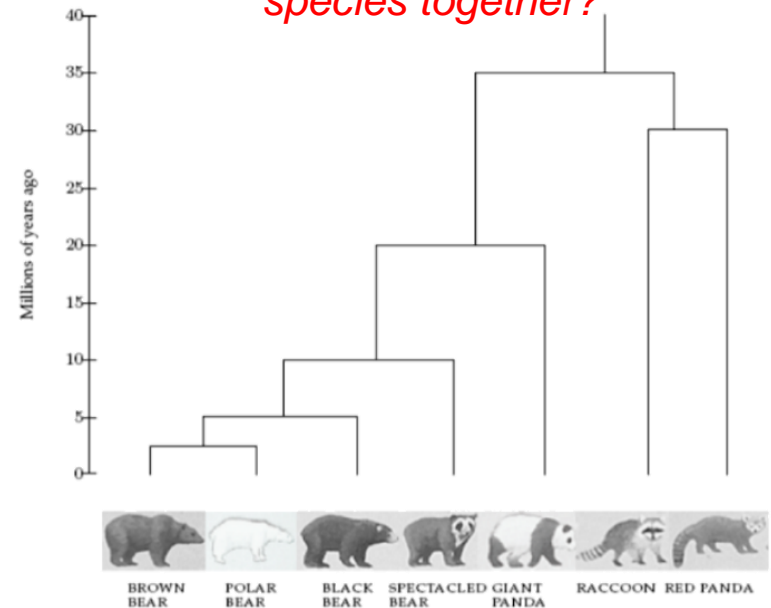
Applications of Hierarchical Clustering

Can we find where a viral outbreak originated?



Tracking Viruses through Phylogenetic Trees

How can we relate different species together?



Construct the phylogenetic tree

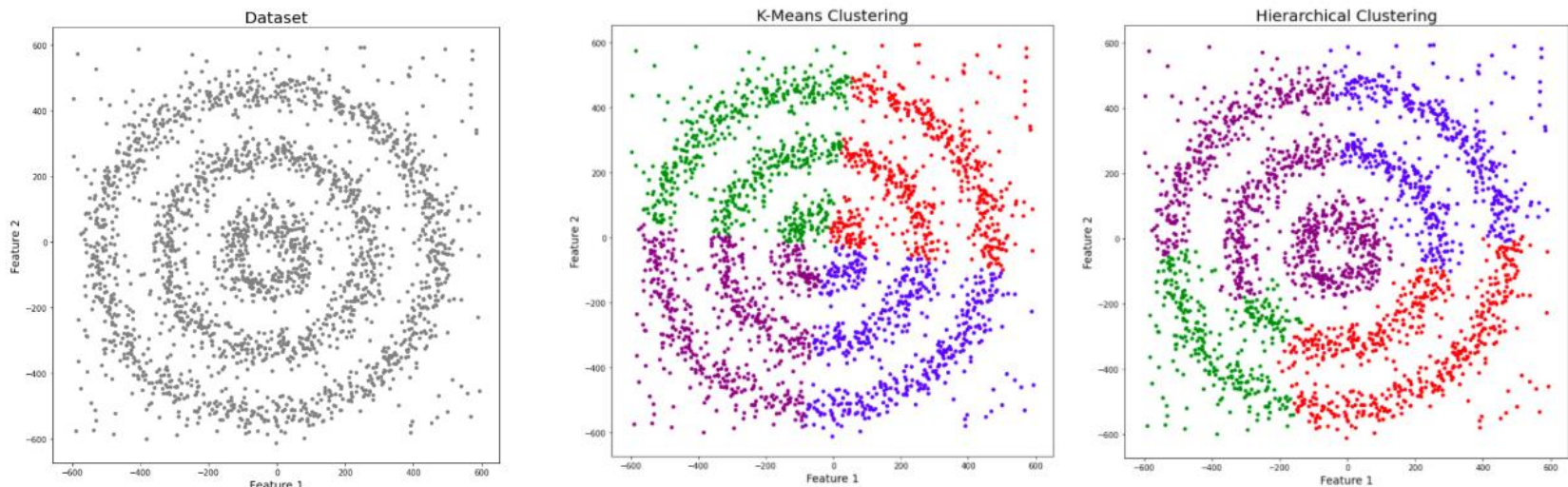
Using DNA sequences and similarities between the DNA sequence, the researchers were able to place the giant pandas closer to bears.

Recent Hierarchical Clustering Methods

- Major weakness of agglomerative clustering methods
 - do not scale well: time complexity of at least $O(n^2)$, where n is the number of total objects
 - can never undo what was done previously
- Integration of hierarchical with distance-based clustering
 - BIRCH (1996): uses CF-tree and incrementally adjusts the quality of sub-clusters
 - ROCK (1999): clustering categorical data by neighbor and link analysis
 - CHAMELEON (1999): hierarchical clustering using dynamic modeling

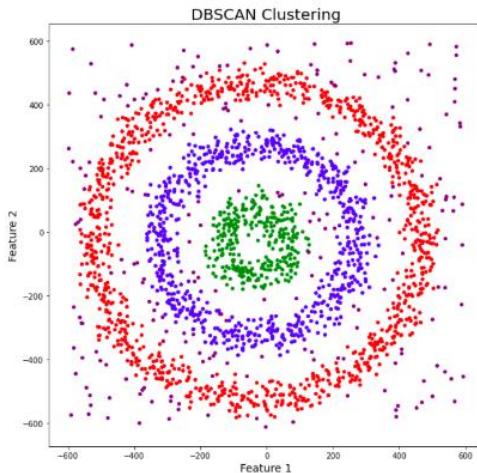
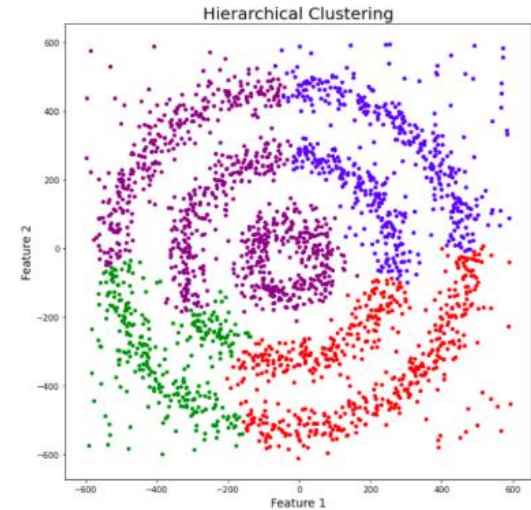
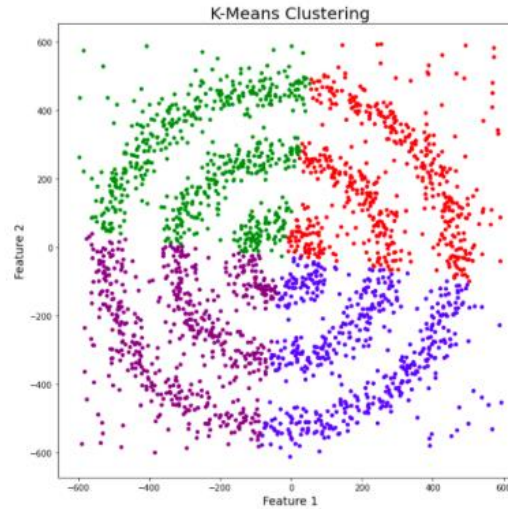
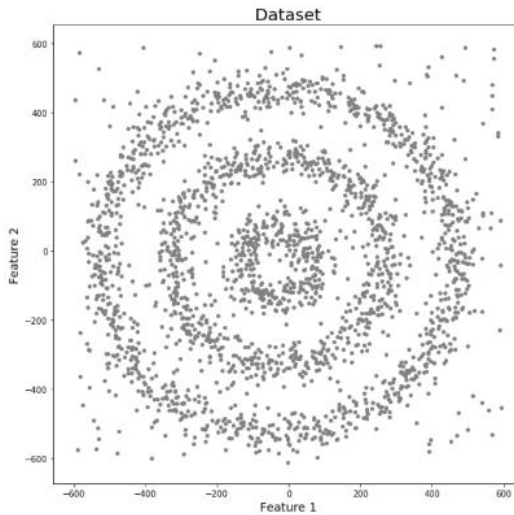
DBSCAN

- **DBSCAN** stands for **D**ensity-**B**ased **S**patial **C**lustering of **A**pplications with **N**oise
- It was proposed by Martin Ester et al. in 1996.
- DBSCAN is a density-based clustering algorithm that works on the assumption that clusters are dense regions in space separated by regions of lower density
- Why DBSCAN is needed



- Purple color – noise
- Both k-means and Hierarchical fail to recognize noise.
- Now lets see the result of DBSCAN

DBSCAN

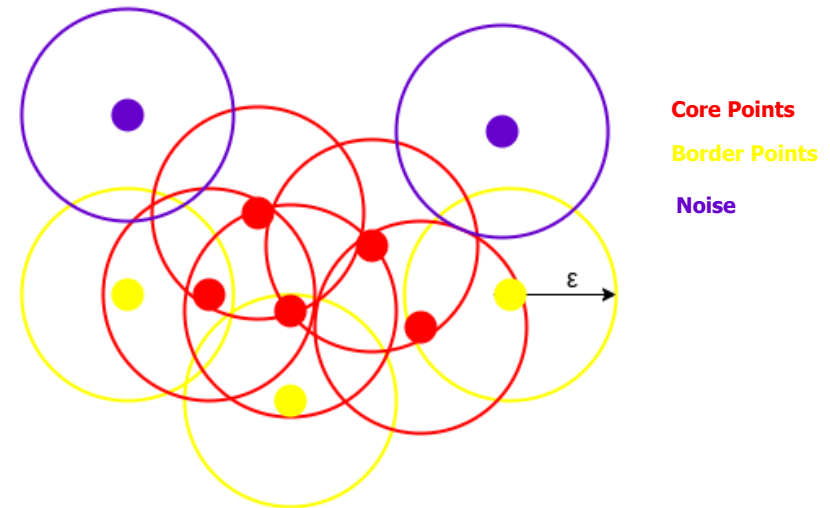


- DBSCAN is not just able to cluster the data points correctly, but it also perfectly detects noise in the dataset.
- It groups 'densely grouped' data points into a single cluster.
- It can identify clusters in large spatial datasets by looking at the local density of the data points.
- The most exciting feature of DBSCAN clustering is that it is robust to outliers.
- It also does not require the number of clusters to be told beforehand, unlike K-Means, where we have to specify the number of centroids.

DBSCAN

DBSCAN requires only two parameters:

1. **Epsilon ϵ** : Epsilon is the radius of the circle to be created around each data point to check the density
2. **minPoints *MinPts*** : minPoints is the minimum number of data points required inside that circle for that data point to be classified as a **Core point**



DBSCAN creates a circle of epsilon radius around every data point and classifies them into Core point, Border point, and Noise.

Core point: if the circle around a data point contains at least 'minPoints' number of points.

Border Point: If the circle around a data point contains less than minPoints,

Noise: if there are no other data points around any data point within epsilon radius.

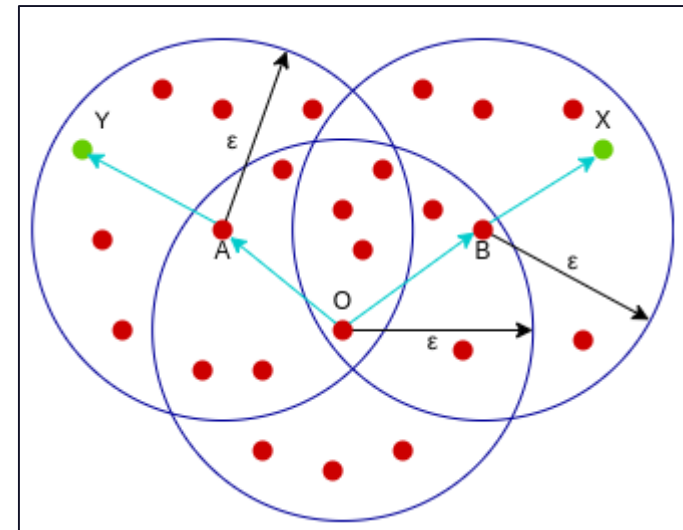
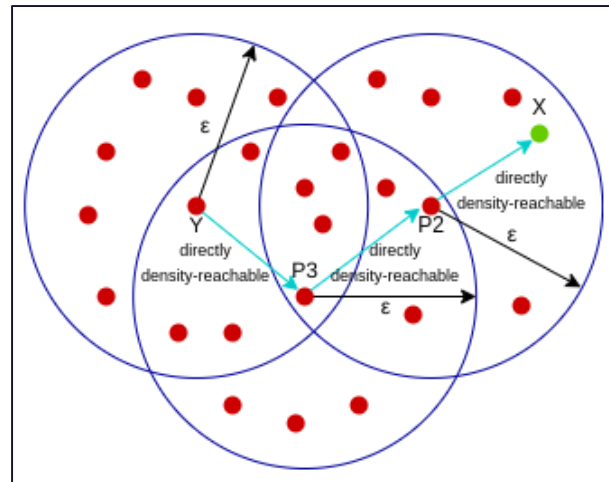
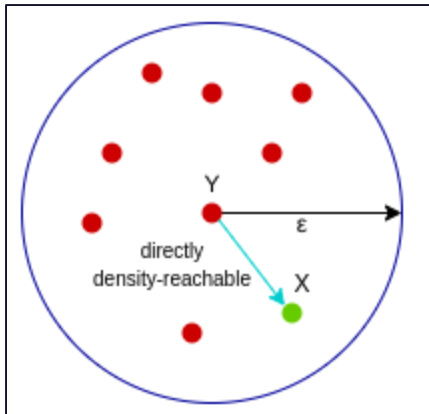
DBSCAN

- Reachability and Connectivity

- Directly Density-Reachable
- Density-Reachable
- Density-Connected

- A point X is directly density-reachable from point Y w.r.t epsilon, minPoints if,

- X belongs to the neighborhood of Y , i.e, $\text{dist}(X, Y) \leq \text{epsilon}$
- Y is a core point



- A point X is density-reachable from point Y w.r.t epsilon, minPoints if

- there is a chain of points $p_1, p_2, p_3, \dots, p_n$ and $p_1=X$ and $p_n=Y$ such that p_{i+1} is directly density-reachable from p_i .

- A point X is density-connected from point Y w.r.t epsilon and minPoints if

- there exists a point O such that both X and Y are density-reachable from O w.r.t to epsilon and minPoints.

DBSCAN technique

Algorithm: DBSCAN: a density-based clustering algorithm.

Input:

- D : a data set containing n objects,
- ϵ : the radius parameter, and
- $MinPts$: the neighborhood density threshold.

Output: A set of density-based clusters.

Method:

- (1) mark all objects as **unvisited**;
- (2) **do**
- (3) randomly select an unvisited object p ;
- (4) mark p as **visited**;
- (5) **if** the ϵ -neighborhood of p has at least $MinPts$ objects
- (6) create a new cluster C , and add p to C ;
- (7) let N be the set of objects in the ϵ -neighborhood of p ;
- (8) **for** each point p' in N
- (9) **if** p' is **unvisited**
- (10) mark p' as **visited**;
- (11) **if** the ϵ -neighborhood of p' has at least $MinPts$ points,
 add those points to N ;
- (12) **if** p' is not yet a member of any cluster, add p' to C ;
- (13) **end for**
- (14) output C ;
- (15) **else** mark p as **noise**;
- (16) **until** no object is **unvisited**;

DBSCAN technique

- Initially, all objects in a given data set D are marked as “unvisited.”
- DBSCAN randomly selects an unvisited object p , marks p as “visited,” and checks whether the ϵ -neighborhood of p contains at least MinPts objects.
- If not, p is marked as a noise point. Otherwise, a new cluster C is created for p , and all the objects in the ϵ -neighborhood of p are added to a candidate set, N .
- DBSCAN iteratively adds to C those objects in N that do not belong to any cluster.
- In this process, for an object p_0 in N that carries the label “unvisited,” DBSCAN marks it as “visited” and checks its ϵ -neighborhood. If the ϵ -neighborhood of p_0 has at least MinPts objects, those objects in the ϵ -neighborhood of p_0 are added to N .
- DBSCAN continues adding objects to C until C can no longer be expanded, that is, N is empty. At this time, cluster C is completed, and thus is output.
- To find the next cluster, DBSCAN randomly selects an unvisited object from the remaining ones. The clustering process continues until all objects are visited.
- If a spatial index is used, the computational complexity of DBSCAN is $O(n \log n)$, where n is the number of database objects. Otherwise, the complexity is $O(n^2)$. With appropriate settings of the user-defined parameters, ϵ and MinPts , the algorithm is effective in finding arbitrary-shaped clusters.

- If Epsilon is 2 and minpoint is 2, what are the clusters that DBScan would discover with the following 8 examples: $A1=(2,10)$, $A2=(2,5)$, $A3=(8,4)$, $A4=(5,8)$, $A5=(7,5)$, $A6=(6,4)$, $A7=(1,2)$, $A8=(4,9)$.

	A1	A2	A3	A4	A5	A6	A7	A8
A1	0	$\sqrt{25}$	$\sqrt{36}$	$\sqrt{13}$	$\sqrt{50}$	$\sqrt{52}$	$\sqrt{65}$	$\sqrt{5}$
A2		0	$\sqrt{37}$	$\sqrt{18}$	$\sqrt{25}$	$\sqrt{17}$	$\sqrt{10}$	$\sqrt{20}$
A3			0	$\sqrt{25}$	$\sqrt{2}$	$\sqrt{2}$	$\sqrt{53}$	$\sqrt{41}$
A4				0	$\sqrt{13}$	$\sqrt{17}$	$\sqrt{52}$	$\sqrt{2}$
A5					0	$\sqrt{2}$	$\sqrt{45}$	$\sqrt{25}$
A6						0	$\sqrt{29}$	$\sqrt{29}$
A7							0	$\sqrt{58}$
A8								0

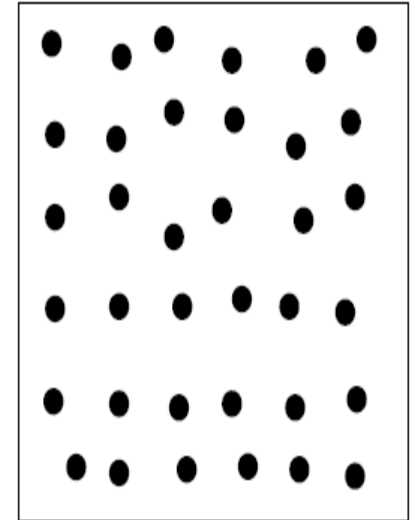
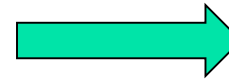
- What is the Epsilon neighborhood of each point?
- $N_2(A1)=\{\}$; $N_2(A2)=\{\}$; $N_2(A3)=\{A5, A6\}$; $N_2(A4)=\{A8\}$; $N_2(A5)=\{A3, A6\}$;
 $N_2(A6)=\{A3, A5\}$; $N_2(A7)=\{\}$; $N_2(A8)=\{A4\}$
- So A1, A2, and A7 are outliers, while we have two clusters $C1=\{A4, A8\}$ and $C2=\{A3, A5, A6\}$

Evaluation of Clustering

- Major clustering evaluation tasks
 - Assessing Clustering Tendency
 - Determining Number of Clusters
 - Measuring Cluster Quality
 - Extrinsic methods
 - Intrinsic methods

Evaluation of Clustering – assessing clustering tendency

- Assessing Clustering Tendency
 - non-random data structure – meaningful clusters
 - Determining if the data set has any non-random data structure.
 - Random data structure means uniform distribution



- Use statistical tests for spatial randomness to measure the probability that the data set is generated by a uniform data distribution.
- Hopkins' statistical testing

Evaluation of Clustering – assessing clustering tendency

Hopkin's Statistics calculation

1. Sample n points, p_1, \dots, p_n , uniformly from D . That is, each point in D has the same probability of being included in this sample. For each point, p_i , we find the nearest neighbor of p_i ($1 \leq i \leq n$) in D , and let x_i be the distance between p_i and its nearest neighbor in D . That is,

$$x_i = \min_{v \in D} \{dist(p_i, v)\}. \quad (10.25)$$

2. Sample n points, q_1, \dots, q_n , uniformly from D . For each q_i ($1 \leq i \leq n$), we find the nearest neighbor of q_i in $D - \{q_i\}$, and let y_i be the distance between q_i and its nearest neighbor in $D - \{q_i\}$. That is,

$$y_i = \min_{v \in D, v \neq q_i} \{dist(q_i, v)\}. \quad (10.26)$$

3. Calculate the Hopkins Statistic, H , as

$$H = \frac{\sum_{i=1}^n y_i}{\sum_{i=1}^n x_i + \sum_{i=1}^n y_i}. \quad (10.27)$$

If D were uniformly distributed, then $\sum_{i=1}^n y_i$ and $\sum_{i=1}^n x_i$ would be close to each other, H would be about 0.5.

if D were highly skewed, then $\sum_{i=1}^n y_i$ would be substantially smaller than $\sum_{i=1}^n x_i$ and thus H would be close to 0

Now,

Null hypothesis – D is uniformly distributed contains no meaningful clusters

Alternate hypothesis – D is not uniformly distributed and contains meaningful clusters

To prove our hypothesis, conduct Hopkin's statistics iteratively with threshold 0.5.

If $H \geq 0.5$, then D is uniformly distributed and

if $H < 0.5$ then D is not uniformly distributed and has statistically significant clusters.

Evaluation of Clustering – determining number of clusters

- **Determining the number of clusters**

- Not very easy – right number is always ambiguous
- Depends on
 - distribution's shape of data
 - Scale of data
 - Required number of clusters by the user

- **Simple method**

- Set the number of clusters to $\sqrt{\frac{n}{2}}$ for a dataset of size n . then each cluster will have $\sqrt{2n}$ points

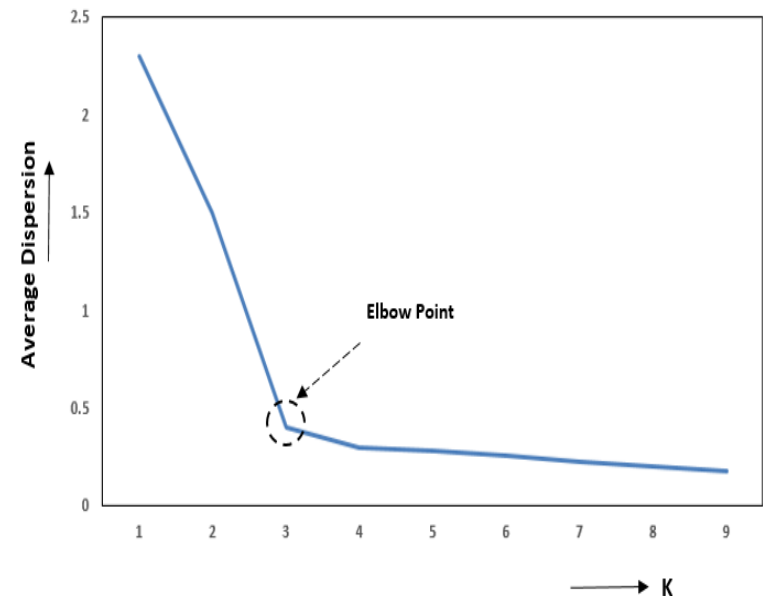
- **Elbow method**

- Increasing the number of clusters can reduce the sum of within-cluster variance
- It will help to create cluster with finer groups i.e. groups that are more similar.
- However, creating too many clusters will reduce this effect of reducing the variance.

Evaluation of Clustering – determining number of clusters

- Selecting the right number of cluster is to use a turning point in the curve of the sum of within-cluster variances with respect to the number of clusters.
- The elbow method plots the value of the cost function (variance) produced by different values of k .
- If k increases, variance will decrease, each cluster will have fewer constituent instances, and the instances will be closer to their respective centroids.
- However, the variance will decline further as k increases.
- The value of k at which improvement in variance declines the most is called the elbow, at which we should stop dividing the data into further clusters.

Elbow Method for selection of optimal “K” clusters



Evaluation of Clustering – measuring clustering quality

- *How good is the clustering generated by a method, and how can we compare the clustering generated by different methods*
- Ground truth - the ideal clustering that is often built using human experts
- Two methods depending on the availability of ground truth
- **Extrinsic method**
 - Supervised
 - ground truth is available i.e. cluster labels are available
 - compare the clustering against the ground truth and measure.
- **Intrinsic method**
 - Unsupervised
 - ground truth is not available
 - evaluate the goodness of a clustering by considering how well the clusters are separated.

Evaluation of Clustering – measuring clustering quality

- **Extrinsic method**

- **Cluster homogeneity**

- Checks for cluster purity – the more pure clusters, the better is the clustering
 - E.g:
 - ground truth – $L_1 \dots L_n$ are the categories of the data for a dataset D
 - Clustering method C_1 places data points from two different categories L_i and L_j in one cluster
 - Clustering method C_2 places data points from two different categories L_i and L_j in different cluster
 - Here C_2 creates pure cluster than C_1
 - So cluster homogeneity score for $C_2 > C_1$

- **Cluster completeness**

- Counter part of cluster homogeneity
 - Cluster completeness requires that for a clustering, if any two objects belong to the same category according to ground truth, then they should be assigned to the same cluster.
 - Clustering method C_1 contains two clusters with data points belonging to same cluster
 - Clustering method C_2 merged the two clusters into one
 - So cluster purity score for $C_2 > C_1$

Evaluation of Clustering – measuring clustering quality

- **Extrinsic method**
 - **Rag bags**
 - are Miscellaneous objects that cannot be merged with other categories
 - The rag bag criterion states that putting a heterogeneous object into a pure cluster should be penalized more than putting it into a rag bag.
 - E.g:
 - Clustering method C1 places all objects belonging to same category in one cluster, except for one object that belongs to label ●
 - Clustering method C2 places object ● in a different cluster C' that contains miscellaneous objects from different categories. So this cluster C' is a noisy cluster i.e. a rag bag.
 - So score of C2 > C1 as per rag bag criterion

Evaluation of Clustering – measuring clustering quality

- **Extrinsic method**

- **Small cluster preservation**

- If a small category is split into small pieces in a clustering, those small pieces may likely become noise and thus the small category cannot be discovered from the clustering.
 - The small cluster preservation criterion states that splitting a small category into pieces is more harmful than splitting a large category into pieces.
 - E.g.,
 - Consider an extreme case. Let D be a data set of $n+2$ objects such that, according to ground truth, n objects, denoted by o_1, \dots, o_n , belong to one category and the other two objects, denoted by o_{n+1}, o_{n+2} , belong to another category.
 - Suppose clustering $C1$ has three clusters, $K1 = \{o_1, \dots, o_n\}$, $K2 = \{o_{n+1}\}$, and $K3 = \{o_{n+2}\}$.
 - Let clustering $C2$ have three clusters, too, namely $K1 = \{o_1, \dots, o_{n-1}\}$, $K2 = \{o_n\}$, and $K3 = \{o_{n+1}, o_{n+2}\}$.
 - Here, $C1$ splits the small category and $C2$ splits the big category.
 - A clustering quality measure Q preserving small clusters should give a higher score to $C2$ as the small category are placed in one cluster and not split into different clusters

-

Evaluation of Clustering

Metrics that satisfy all the four criteria of extrinsic methods:

BCubed precision: how many other objects in the same cluster belong to the same category as the object

BCubed recall: how many objects of the same category are assigned to the same cluster

Formally, let $D = \{o_1, \dots, o_n\}$ be a set of objects, and \mathcal{C} be a clustering on D . Let $L(o_i)$ ($1 \leq i \leq n$) be the category of o_i given by ground truth, and $C(o_i)$ be the *cluster_ID* of o_i in \mathcal{C} . Then, for two objects, o_i and o_j , ($1 \leq i, j \leq n, i \neq j$), the *correctness* of the relation between o_i and o_j in clustering \mathcal{C} is given by

$$\text{Correctness}(o_i, o_j) = \begin{cases} 1 & \text{if } L(o_i) = L(o_j) \Leftrightarrow C(o_i) = C(o_j) \\ 0 & \text{otherwise.} \end{cases} \quad (10.28)$$

BCubed precision is defined as

$$\text{Precision BCubed} = \frac{\sum_{i=1}^n \frac{\sum_{o_j: i \neq j, C(o_i) = C(o_j)} \text{Correctness}(o_i, o_j)}{\|\{o_j | i \neq j, C(o_i) = C(o_j)\}\|}}{n}. \quad (10.29)$$

BCubed recall is defined as

$$\text{Recall BCubed} = \frac{\sum_{i=1}^n \frac{\sum_{o_j: i \neq j, L(o_i) = L(o_j)} \text{Correctness}(o_i, o_j)}{\|\{o_j | i \neq j, L(o_i) = L(o_j)\}\|}}{n}. \quad (10.30)$$

Evaluation of Clustering – Intrinsic method

- **Intrinsic method**

- Unsupervised – ground truth is not available
- evaluate a clustering by examining how well the clusters are separated and how compact the clusters are
- **Silhouette coefficient**
 - For a data set, D , of n objects, suppose D is partitioned into k clusters, C_1, \dots, C_k .
 - For each object $o \in D$,
 - $a(o)$ as the average distance between o and all other objects in the cluster to which o belongs i.e. reflects the compactness of the cluster to which o belongs.
 - Similarly, $b(o)$ is the minimum average distance from o to all clusters to which o does not belong i.e. captures the degree to which o is separated from other clusters.

$$a(o) = \frac{\sum_{o' \in C_i, o' \neq o} \text{dist}(o, o')}{|C_i| - 1}$$

the
the
containing o is

$$b(o) = \min_{C_j: 1 \leq j \leq k, j \neq i} \left\{ \frac{\sum_{o' \in C_j} \text{dist}(o, o')}{|C_j|} \right\}$$

the
the

The silhouette coefficient of o is then defined as

$$s(o) = \frac{b(o) - a(o)}{\max\{a(o), b(o)\}}$$

Evaluation of Clustering – Intrinsic method

- However, when the silhouette coefficient value is negative (i.e., $b(o) < a(o)$), this means that, in expectation, o is closer to the objects in another cluster than to the objects in the same cluster as o .
- In many cases, this is a bad situation and should be avoided.
- To measure a cluster's fitness within a clustering, we can compute the average silhouette coefficient value of all objects in the cluster. To measure the quality of a clustering, we can use the average silhouette coefficient value of all objects in the data set.
- The silhouette coefficient and other intrinsic measures can also be used in the elbow method to heuristically derive the number of clusters in a data set by replacing the sum of within-cluster variances.