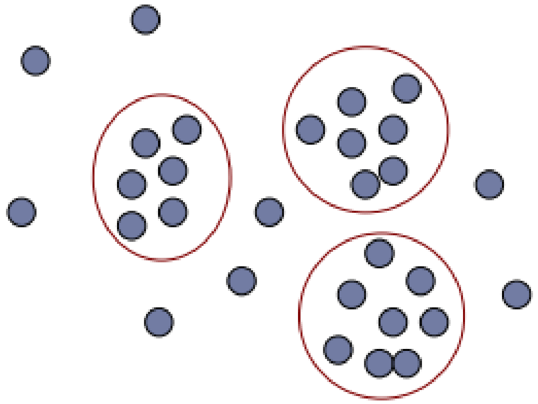
Cluster Analysis

Jiaying Liu G45268292

Hao Wu G36059463

What is Cluster Analysis

- Unsupervised learning (i.e., class label is unknown)
- Group data to form new categories (i.e., clusters), e.g., cluster houses to find distribution patterns
- Principle: Maximizing intra-class similarity & minimizing interclass similarity



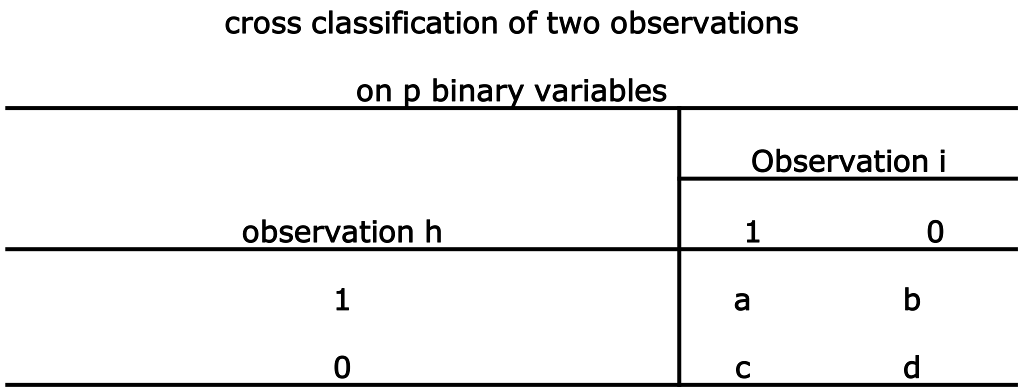
- Typical Applications: “Market basket data”, “Amazon recommends products”, “biology clustering” etc.

Supervised VS. Unsupervised Learning

- Supervised: Dataset with pre-defined classification that we can use this to supervised learn the dataset and create the model
- Example: discriminant analysis, logistic regression
- Unsupervised: The classification categories are unknown, but features are observed that relate to the unobserved categories.
- Example: cluster method

Measuring Dissimilarity Between Observations

- For **categorical data**, vectors of observations on p binary variables.
- P=a+b+c+d
- Similarity (matching index): $\frac{a + d}{a + b + c + d}$
- Dissimilarity: $\frac{b + c}{a + b + c + d}$



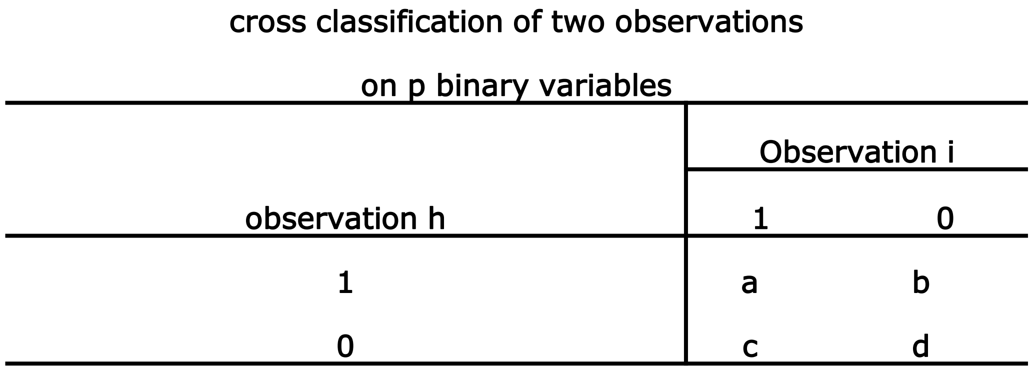
cross classification of two observations

on p binary variables

	Observation i	
observation h	1	0
1	a	b
0	c	d

Measuring Dissimilarity Between Observations (cont.)

- In special applications, value 1 is more common (or meaningful) than value 0.
- Jaccard index: $\frac{a}{a+b+c}$
- Dissimilarity: $\frac{b+c}{a+b+c}$



cross classification of two observations

on p binary variables

	Observation i	
observation h	1	0
1	a	b
0	c	d

Measuring Dissimilarity Between Observations (cont.)

- For **continuous data**, we usually calculate the distance between sample points.
- Treat data as point (or vector) in the dimensional space.
- Smaller distance, larger similarity.

- **Minkowski diatance:**
- For $p = [p_1, p_2, \dots, p_m]$ and $q = [q_1, q_2, \dots, q_m]$
- $d_x(p, q) = (\sum_{i=1}^{m} |p_i - q_i|^x)^{1/x}$, (x>0)

Measuring Dissimilarity Between Observations (cont.)

- Minkowski diatance: $d_x(p,q) = \left(\sum_{i=1}^{m} |p_i - q_i|^x\right)^{1/x}$, (x>0)
- x=1,
- $d_1(p,q) = \sum_{i=1}^{m} |p_i - q_i|$ **Hamming distance**
- X=2,
- $d_2(p,q) = \sqrt{\sum_{i=1}^{m} |p_i - q_i|^2}$ **Euclidean distance**
- X= ∞,
- $d_\infty(p,q) = \max_{1 \le i \le m} |p_i - q_i|$ **Chebyshev distance**

May be effected by the unit of measurement, require normalization before use!

Measuring Dissimilarity Between Observations (cont.)

- **Canberra diatance:** $d_{canb}(p,q) = \sum_{i=1}^{m} \frac{|p_i - q_i|}{|p_i| + |q_i|}$
- Overcome the influence of measurement unit
- When coordinates of two points are both close to 0, Canberra distance is sensitive with tiny changes.

- Besides distance, we can also use coefficient (like correlation coefficient) to measure similarity.

Clustering Categories

- Partitioning Methods
  - Construct K partitions of the data
    - K-mean Clustering
    - K-medoid Clustering
- Hierarchical Methods
  - Creates a hierarchical decomposition of the data
    - Bottom – up / agglomerative
    - Top-down / divisive

Partitioning Methods: The Principle

- Given
  - A data set of n objects
  - K the number of clusters to form

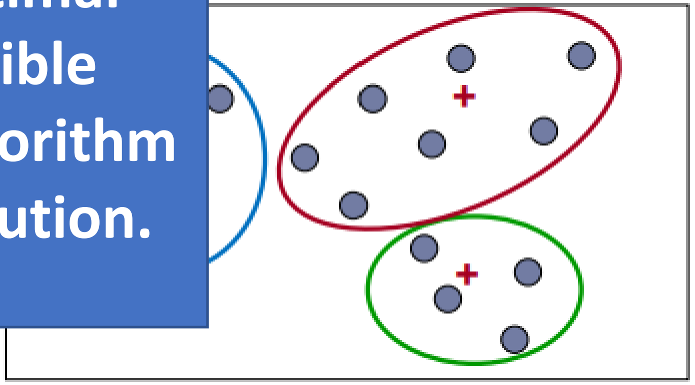
- Organize the objects into K partitions(k<=n) where each partition represents a cluster

- The cluster are formed to optimize an objective partitioning criterion
  - Objects within a cluster are similar
  - Objects of different clusters are dissimilar

Partitions----K-mean clustering

- Idea: Find th allest total dissimilarity

- The algorith partition tha function



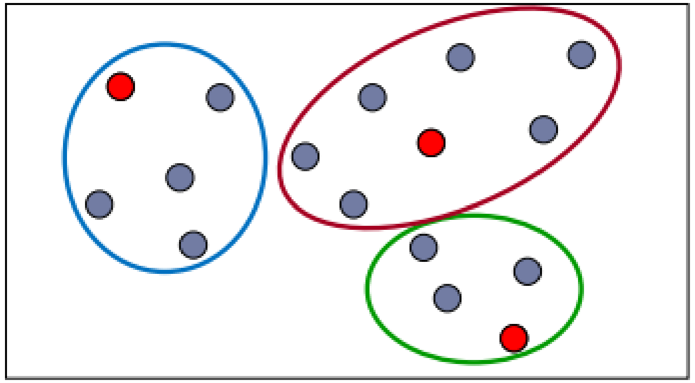
Actually it is not easy to minimize the square-error function, to find the optimal solution, we need to find all the possible cluster. So k mean use the greedy algorithm to find the literately approximant solution.

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

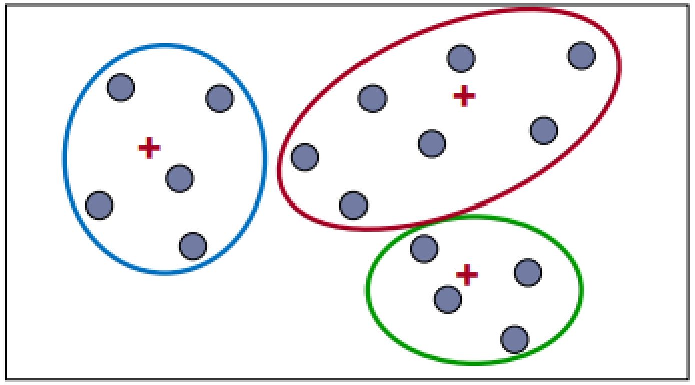
- It works well when the clusters are compact clouds that are rather well separated from one another

Basic Process of K-mean Clustering

- 1. Arbitrary choose k objects from D as in initial cluster centers

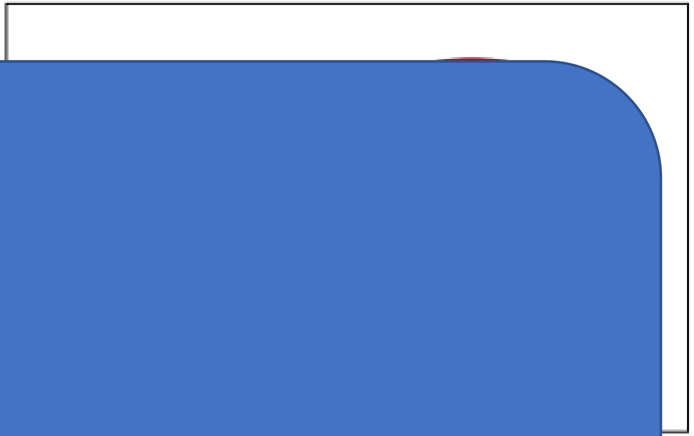


- 2. Assign each object to the most similar cluster based on the mean value of the objects in the cluster



Process of K-mean clustering

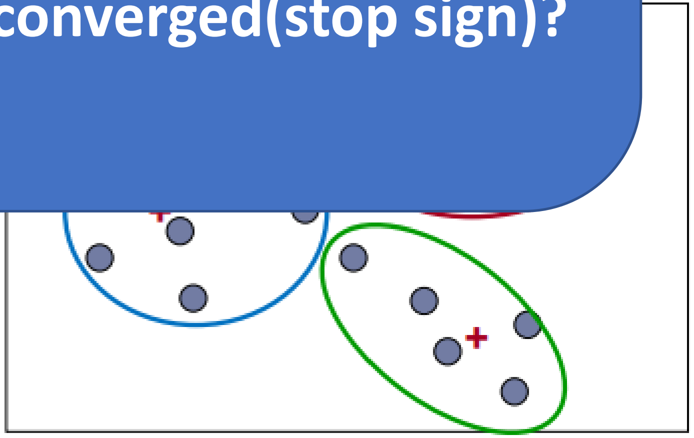
- 3. U



Question:

1. How to find the initial value?
2. How to cluster and update ?
3. How to decide that the algorithm is converged(stop sign)?

- 4.If



```
• Input D={$x_{1,}$ $x_2$, $x_3$, …, $x_n$}; # of Cluster $k$.
• { Randomly choose $k$ points as initial $\mu = \{\mu_1, \mu_2, \mu_3, \dots, \mu_k\}$
   Repeat
       let $C_i = \emptyset$ $(1 \leq i \leq k$ )
       for j = 1,2,…,n do
           calculate each points $x_j$ distance with each $\mu_i$: $d_{ij} = \left|\left|x_j - \mu_i\right|\right|_2$
           find the minimum distance and set it to that cluster
       end for
       for i=1,2,…k do
           calculate the new $\mu_i' = \frac{1}{|C_i|}\sum_{x \in C_i} x$;
           if $\mu_i' \neq \mu_i$ then
               update $\mu_i = \mu_i'$
           else
               keep the $\mu_i$
           end if
       end for
   until $\mu_i$ nerve update
Output cluster= C= $\{C_1, C, C_3, \dots, C_k\}$
```

K-Mean Properties

Advantages

- K-means is relatively scalable and efficient in processing large data sets
- The computational complexity of the algorithm is O(nkt)

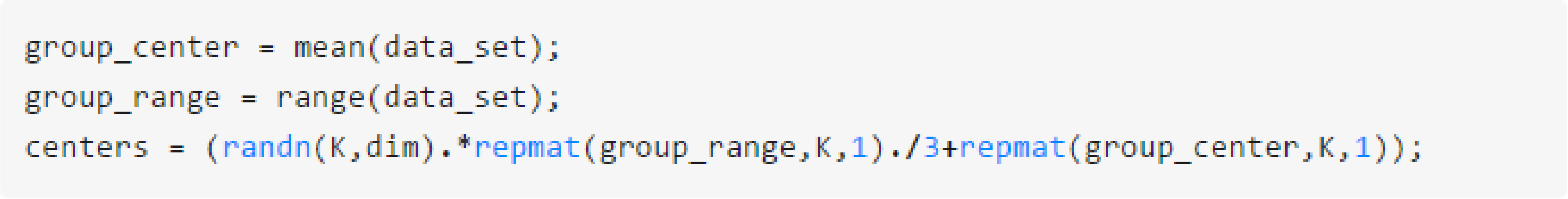
Disadvantage

- **it is applicable only when the mean is defined (Cannot apply for categorical data)**
- Users need to specify k
- K-means is not suitable for discovering clusters with nonconvex shapes or clusters of very different size
- It is sensitive to noise and outlier data points (can influence the mean value)

Optimize the algorithm

- How to find the K?
  - Plots the value of the clustering criterion (cost function) against k, looking for a natural break point where this changes substantially
  - Plot a summary against k such as the probability that the dissimilarity for a with-in cluster pair is smaller than the dissimilarity for a between-cluster pair.

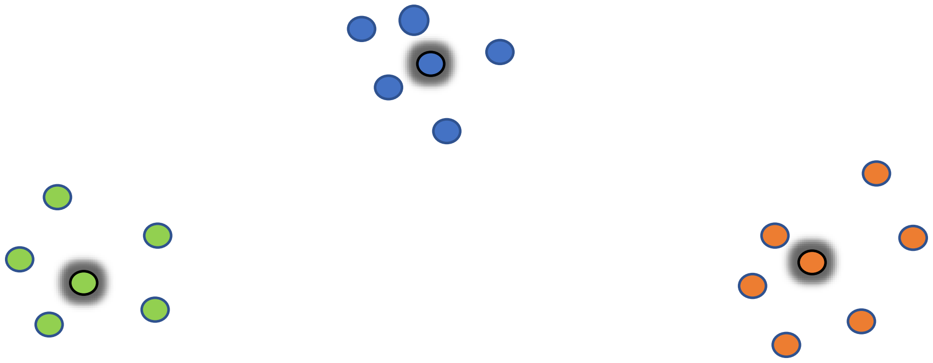
- What initial value can make algorithm converge more quickly?
    - Even we say we can randomly choose k point and since the algorithm will update through each iteration. In the end, the mean will move and eventually go to the right place;
    - But image: what is it take a lot of time? what if it vibrate? What if the point we choose is outlier????
    - First estimate the center of whole dataset and create k point which are the value of scale multiple random normal value like:



```
group_center = mean(data_set);
group_range = range(data_set);
centers = (randn(K,dim).*repmat(group_range,K,1)./3+repmat(group_center,K,1));
```

Partitions----K-medoid clustering (PAM)

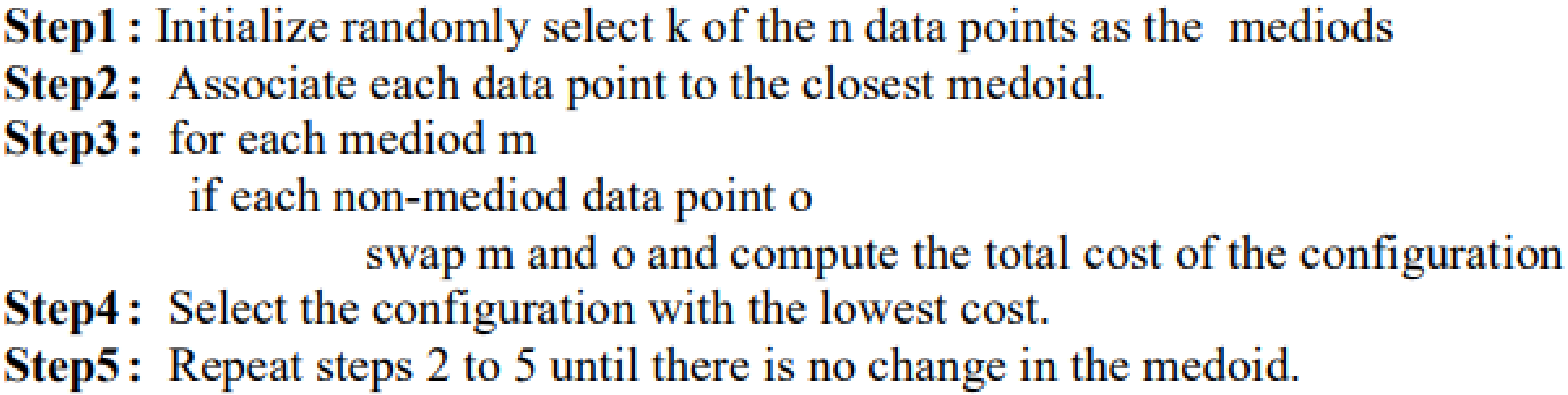
- Medoid: The medoid of a cluster is the truly observation with smallest total dissimilarity to the other points in the cluster



- The goal of k-medoid clustering is to minimizes the sum of the dissimilarities between each object and its corresponding reference point (medoid)

$$E = \sum_{i-1}^{k} \sum_{p \in C_i} | p - o_i |$$

Process of K-medoid clustering

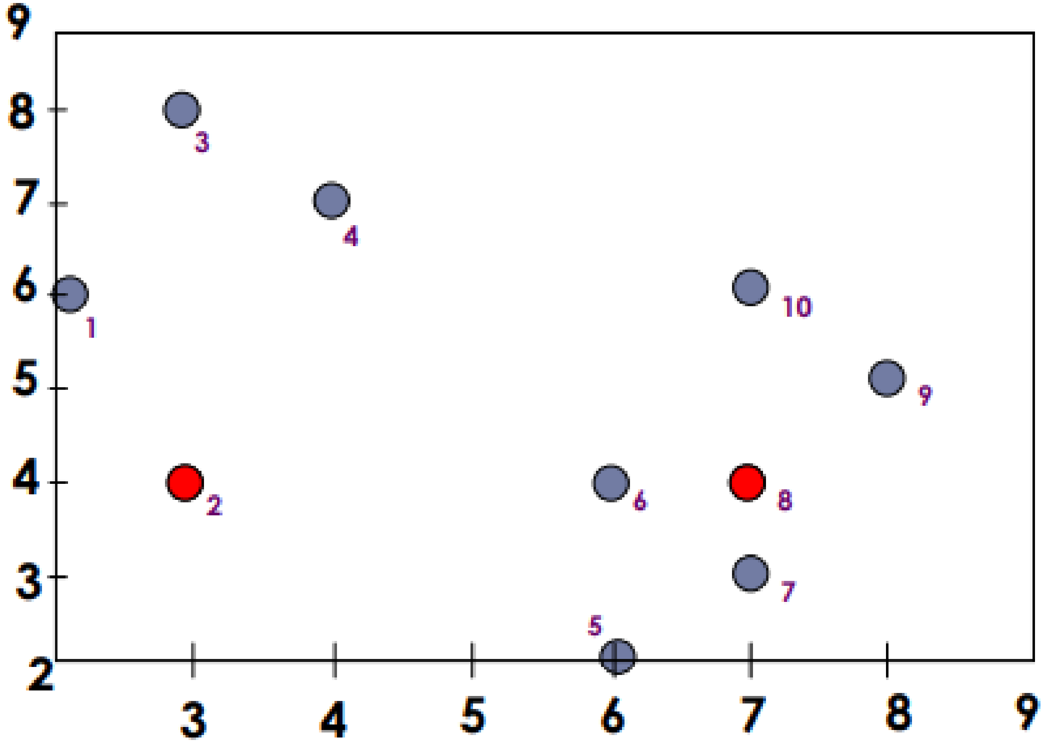


```
Step1 : Initialize randomly select k of the n data points as the  mediods
Step2 :  Associate each data point to the closest medoid.
Step3 :  for each mediod m
           if each non-mediod data point o
                   swap m and o and compute the total cost of the configuration
Step4 :  Select the configuration with the lowest cost.
Step5 :  Repeat steps 2 to 5 until there is no change in the medoid.
```

Example of k-medoid clustering

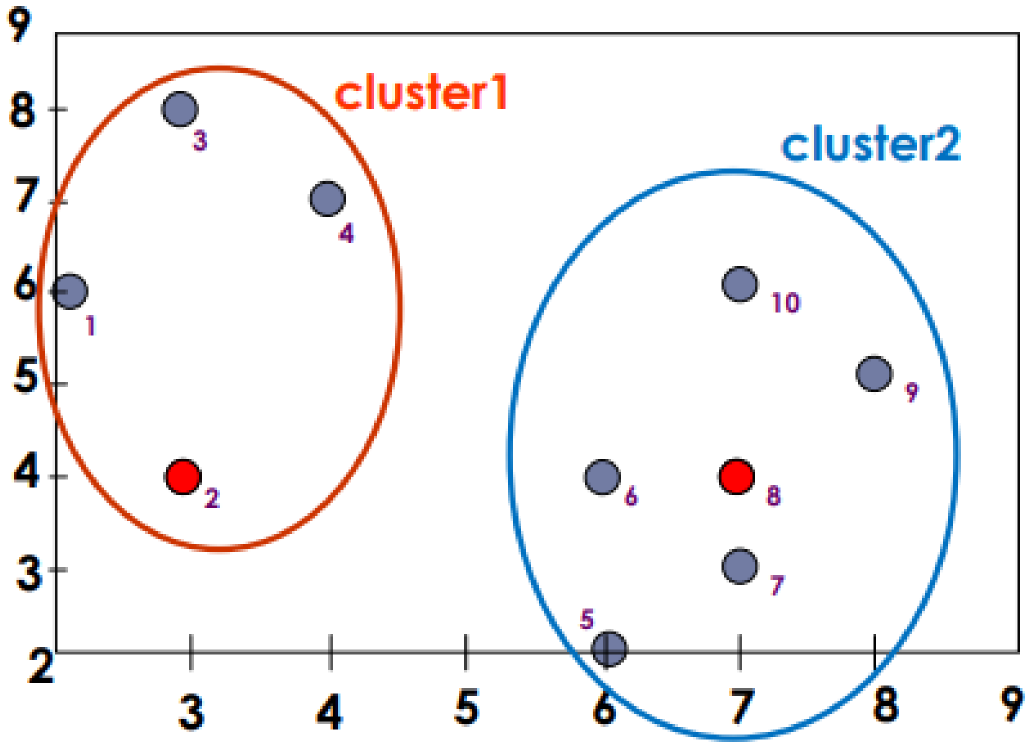
Data objects

	A_1	A_2
O_1	2	6
O_2	3	4
O_3	3	8
O_4	4	7
O_5	6	2
O_6	6	4
O_7	7	3
O_8	7	4
O_9	8	5
O_{10}	7	6



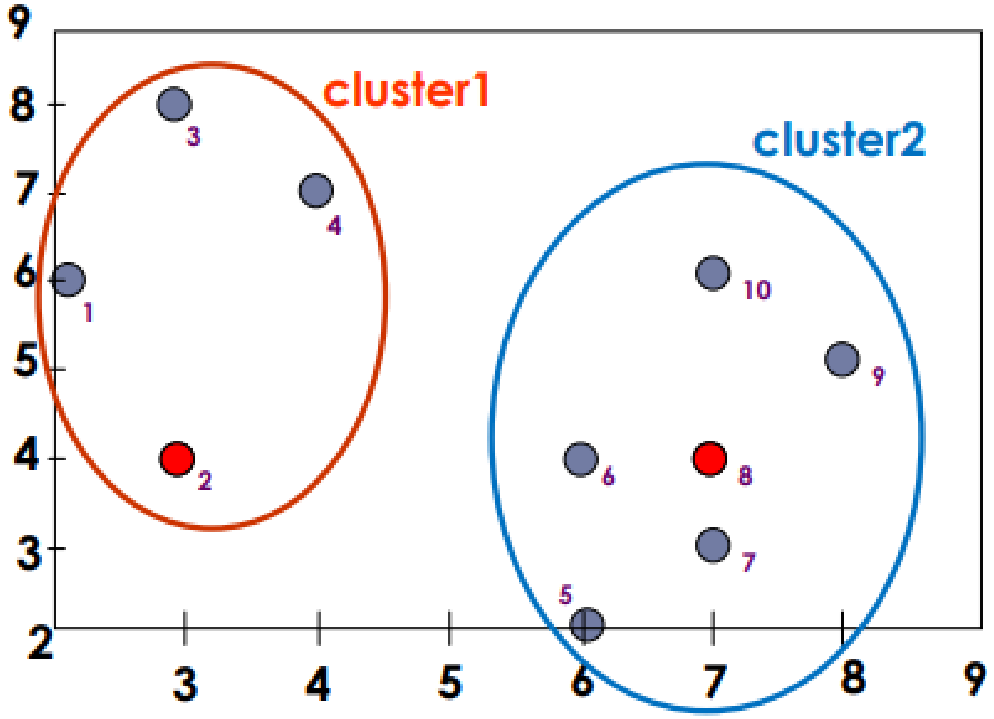
Goal: create two cluster
Choose randomly two medoids
O2(3,4), O8(7,4)

Example of k-medoid clustering



- Assign each object to the closest representative object
- Using distance to form the following clusters
- Cluster1= {O1,O2,O3,O4}
- Cluster2={O5,O6,O7,O8,O9,O10)

Example of k-medoid clustering



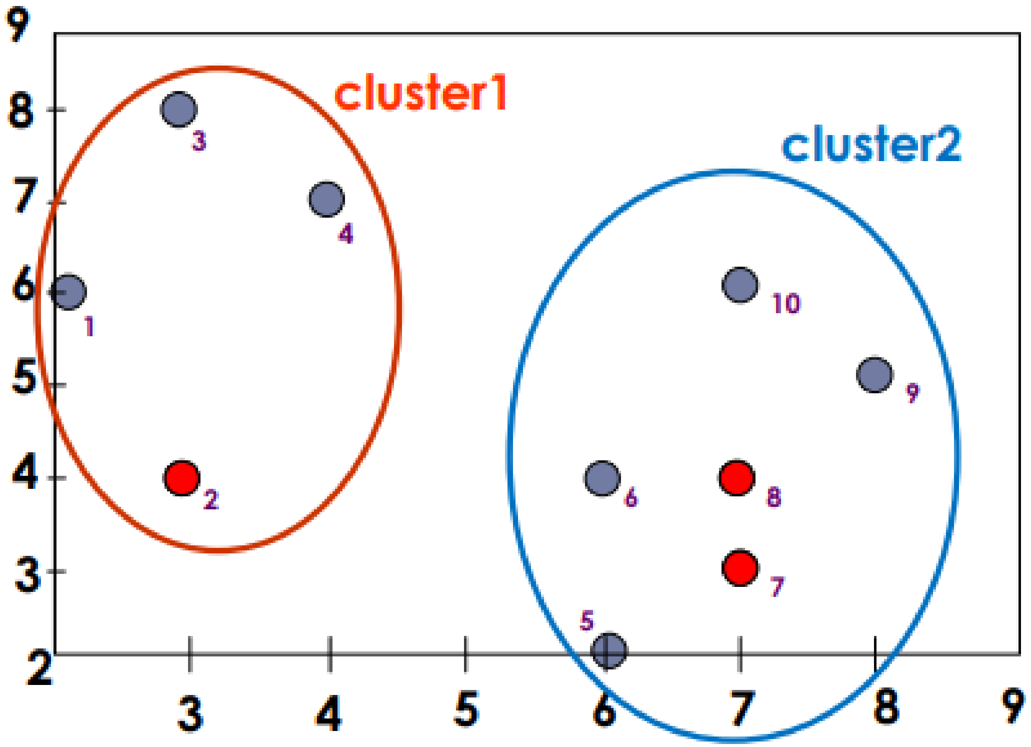
- Compute the absolute error function [for the set of Medoids (O2,O8)]

$$E=\sum_{i-1}^{k}\sum_{p\in C_i}|p-o_i|=|o_1-o_2|+|o_3-o_2|+|o_4-o_2|$$

$$+|o_5-o_8|+|o_6-o_8|+|o_7-o_8|+|o_9-o_8|+|o_{10}-o_8|$$

$$E=(3+4+4)+(3+1+1+2+2)=20$$

Example of k-medoid clustering



Choose a random object O7
Swap O8 and O7
Compute the absolute error function [for the set of Medoids (O2,O7)]

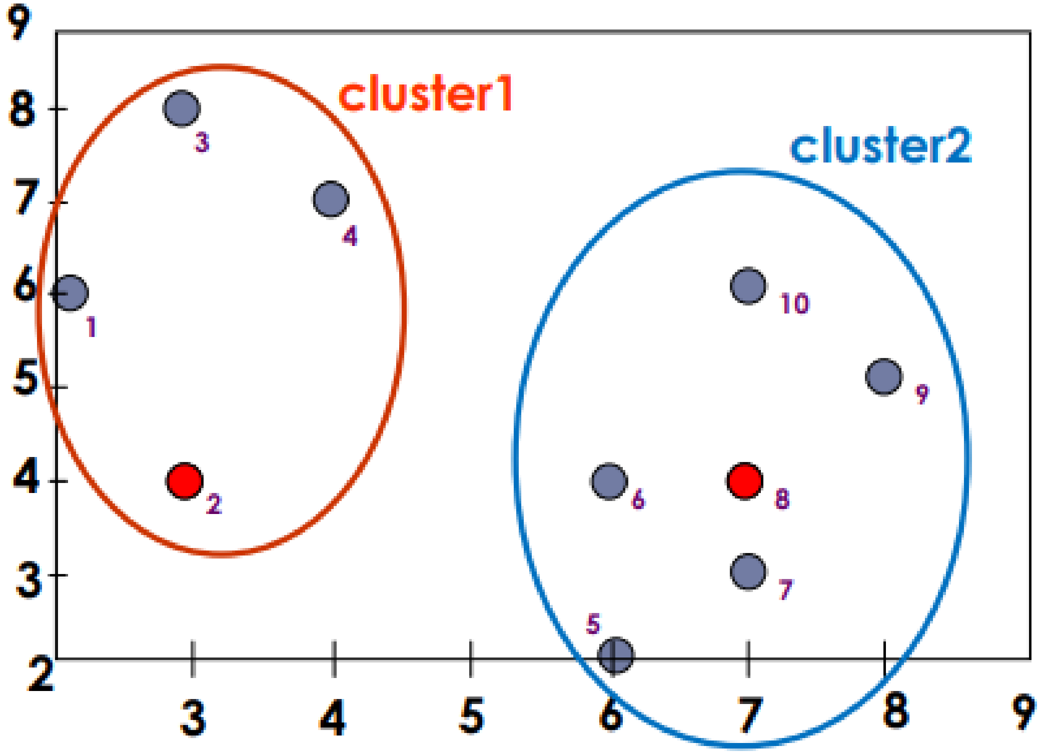
$$E = (3 + 4 + 4) + (2 + 2 + 1 + 3 + 3) = 22$$

Compute the cost function
Absolute error [for O2,O7] – Absolute error [O2,O8]

$$S = 22 - 20$$

S>0 It is a bad idea to replace O8 by O7

Example of k-medoid clustering



- Since there is no change in the medoid set, the algorithm ends here. Hence the clusters obtained finally are
- Cluster1= {O1,O2,O3,O4}
- Cluster2={O5,O6,O7,O8,O9,O10)

K-medoids properties(k-medoids vs. k-means)

- Advantages
  - K-Medoids method is more robust than k-Means in the presence of noise and outliers
- Disadvantages
  - K-Medoids is more cost that the k-Means method
  - Like k-means, k-medoids requires the user to specify k
  - It does not scale well for large data sets
  - (CLARA[Clustering Large Application],CLARANS[Clustering Large Application based upon Randomized Search)

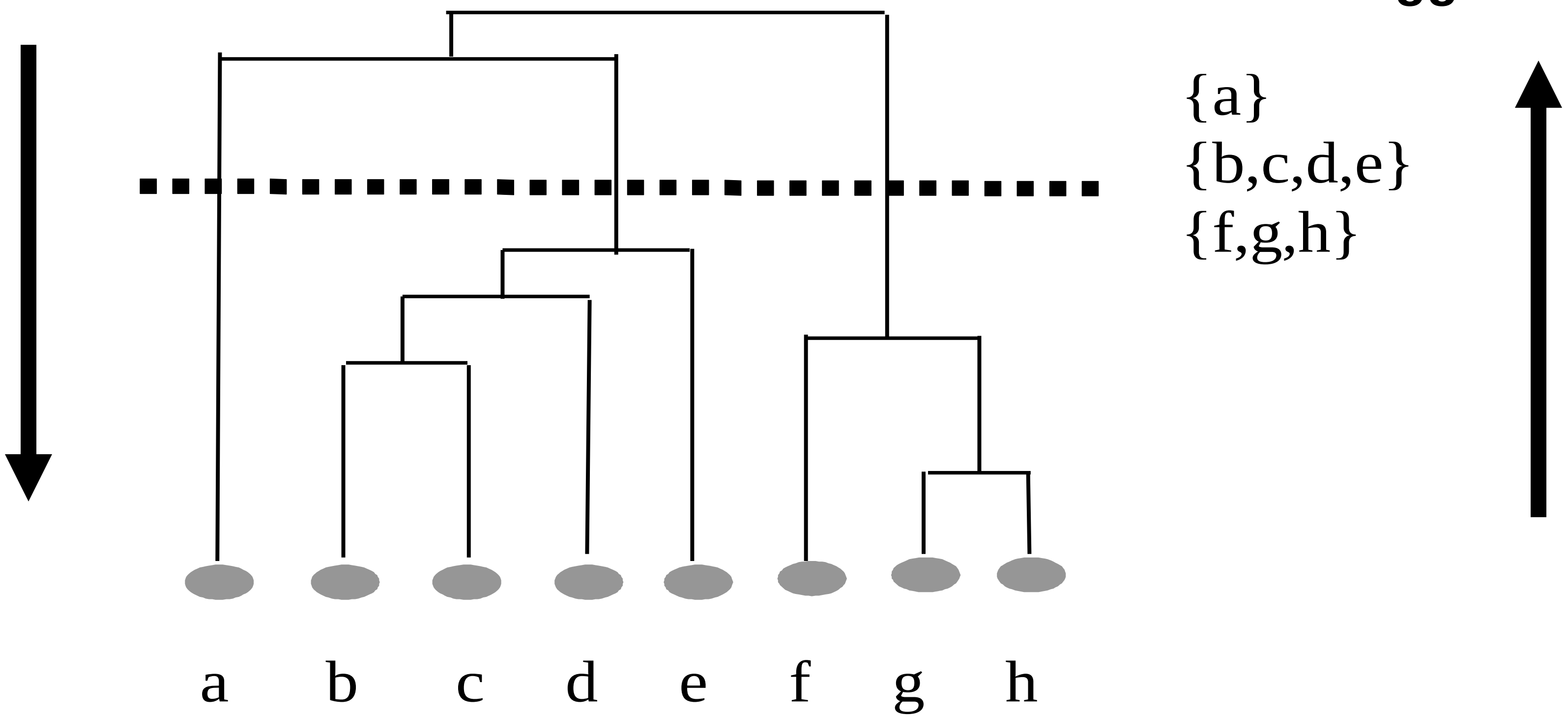
Hierarchies clustering

- HC provides graphical illustration of relationships between the data in

the form of a dendrogram (binary tree).

- Two approaches: agglomerative & divisive
- Agglomerative / bottom-up method starts with each object in the data forming its own cluster, and then successively merges the clusters until one large cluster is formed, which encompasses the entire dataset.
- Divisive / top-down method starts by considering the entire data as one cluster and then splits up the cluster(s) until each object forms its own cluster.

Dendrogram



Top –down / divisive

Bottom-up / agglomerative

{a}
{b,c,d,e}
{f,g,h}

Procedure of Agglomerative Clustering

Given : a data set and the distance function

1. start with “N “ clusters by assigning each pattern to a separate cluster

2. proceed with this initial configuration of the clusters and merge the clusters that are the closest. In other words, if S and T are the two clusters being recognized as the closest, form a single cluster {S, T} and reduce the number of clusters by one

Set a threshold. If the smallest distance less than threshold, stop iretation.

3 repeat step 2 until a minimal number of the clusters has been reached.

Result : clusters of data (partition)

Distance Between Clusters

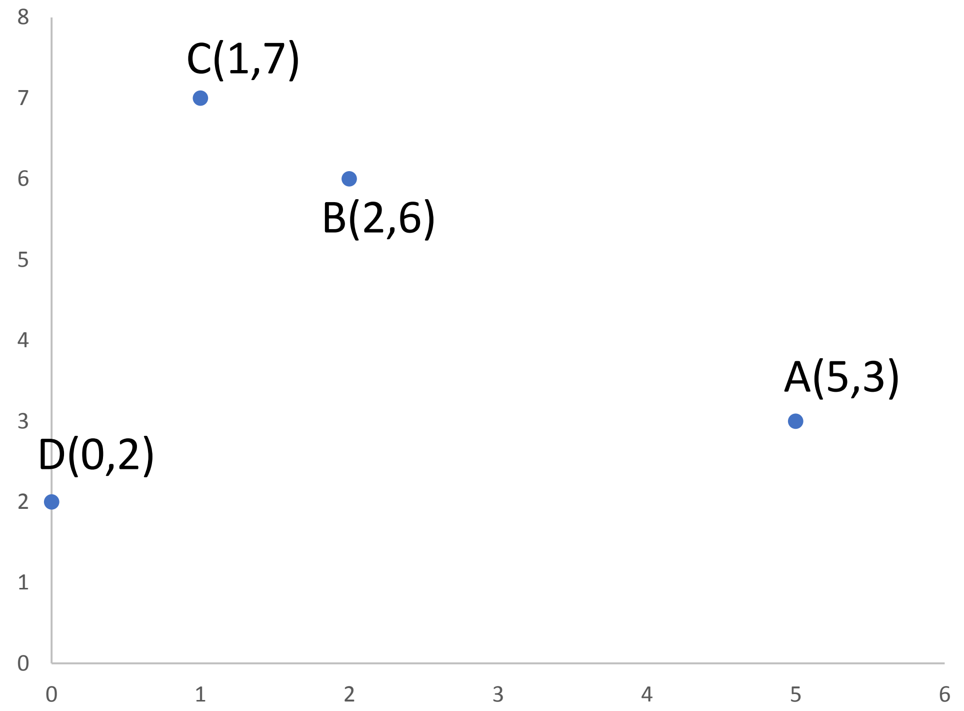
$$\text{Single linkage method}: \|T-S\| = \min_{\substack{x\in T\\ y\in S}} \|x-y\|$$

$$\text{complete linkage}: \|T-S\| = \max_{\substack{x\in T\\ y\in S}} \|x-y\|$$

$$\text{average linkage}: \|T\text{-}S\| = \frac{1}{card(S)card(T)} \sum_{\substack{x\in T\\ y\in S}} \|x-y\|$$

Example

- Suppose 4 data: A(5,3), B(2,6), C(1,7), D(0,2)
- We want to divide data into two clusters.
- Use agglomerative clustering.
- Measure dissimilarity between observations with Euclidean distance.
- Measure distance between clusters with average linkage.

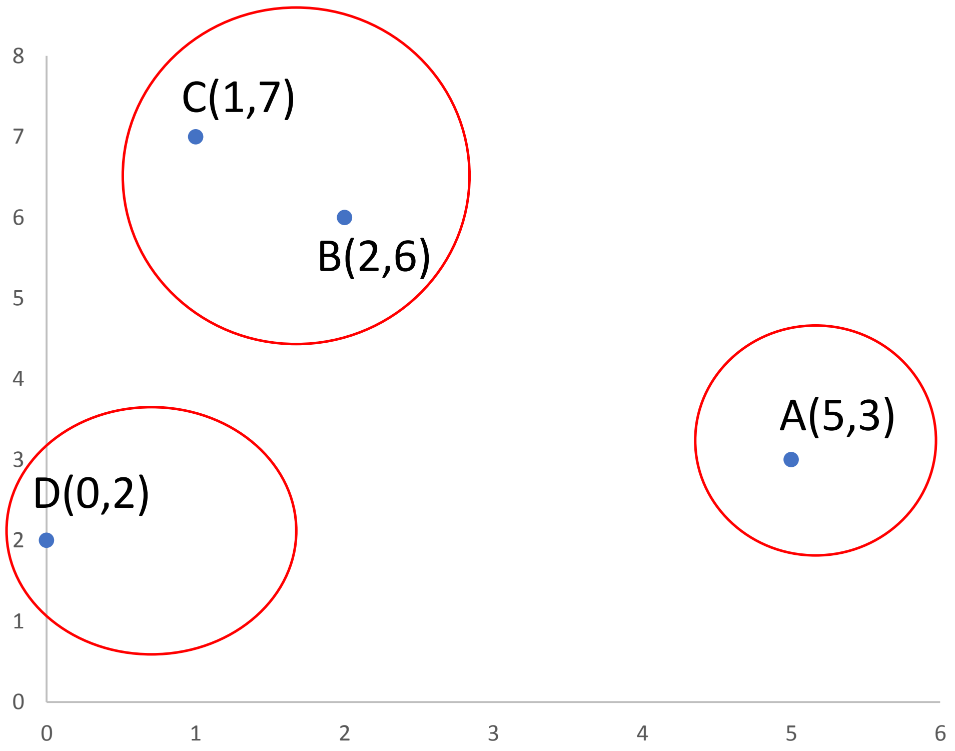


Example

- At initial stage: treat each observation as a single cluster.
- Calculate dissimilarity:

	A	B	C	D
A		4.24	5.66	5.10
B	4.24		**1.41**	4.47
C	5.66	**1.41**		5.10
D	5.10	4.47	5.10	

- Combine the clusters with smallest distance as one cluster.
- Now 3 clusters: {A}, {B,C}, {D}



Example

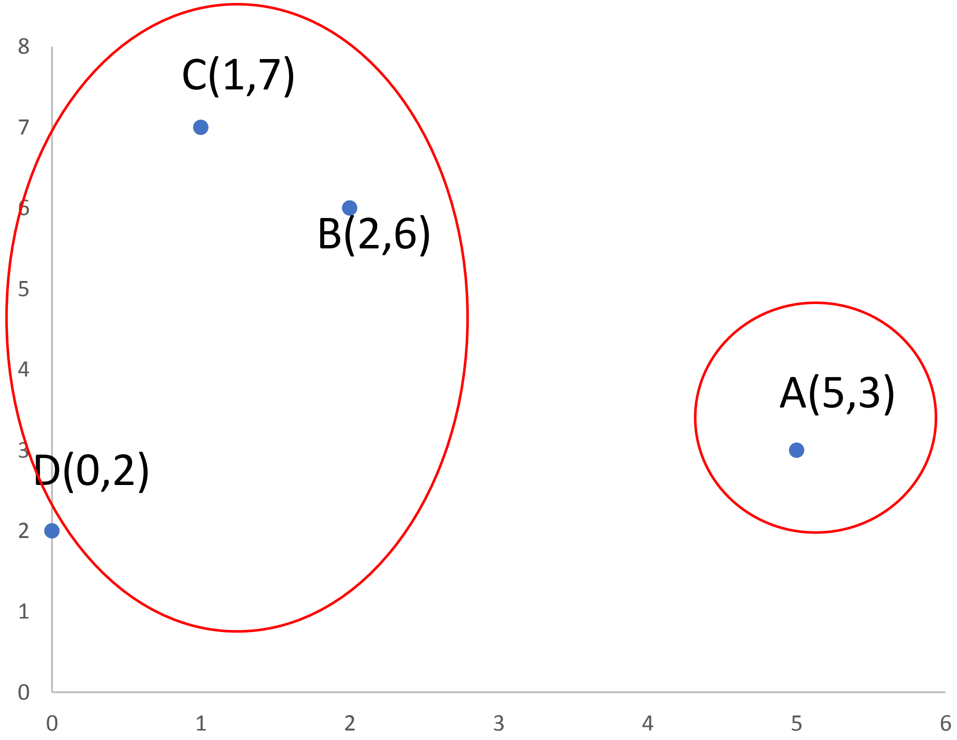
- Calculate cluster distance:

	A	B&C	D
A		4.95	5.10
B&C	4.95		4.79
D	5.10	4.79	

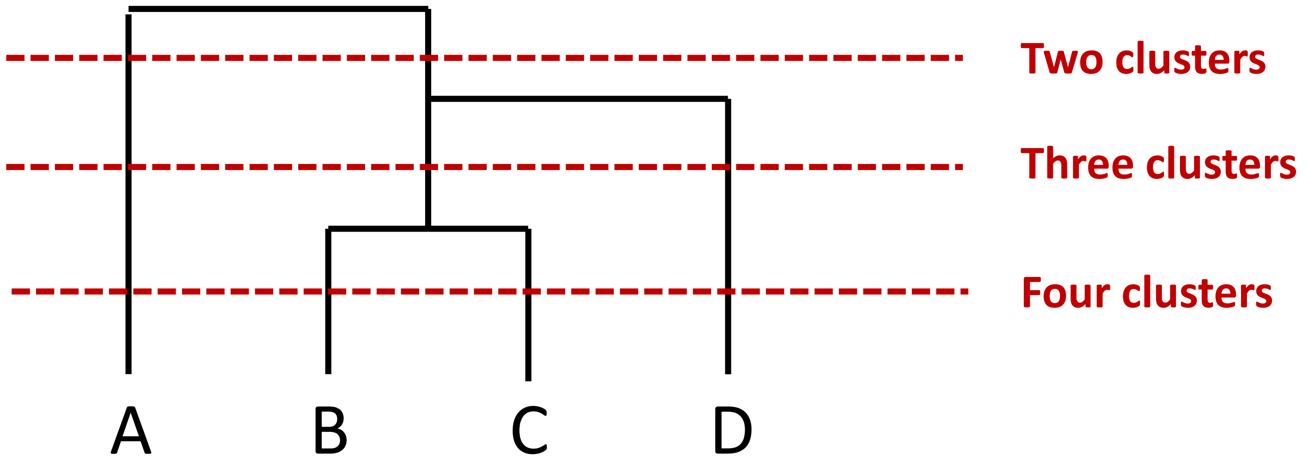
- Combine the clusters with smallest distance as one cluster.
- Now 2 clusters: {A}, {B,C,D}

Average linkage between {A} and {B,C}

$$=\frac{1}{2}[\sqrt{(A-B)^2}+\sqrt{(A-C)^2}]$$



Example



Reference

- https://cran.r-project.org/web/packages/kmed/vignettes/kmedoid.html
- https://www.mathworks.com/help/stats/kmedoids.html
- http://www.sthda.com/english/wiki/print.php?id=236
- https://www.datanovia.com/en/lessons/clara-in-r-clustering-large-applications/
- https://en.wikibooks.org/wiki/Data_Mining_Algorithms_In_R/Clustering/CLARA
- http://www.sthda.com/english/wiki/print.php?id=239
- https://cran.r-project.org/web/packages/cluster/cluster.pdf
- https://cran.r-project.org/web/packages/kmed/kmed.pdf
- https://cran.r-project.org/web/packages/kmed/vignettes/kmedoid.html