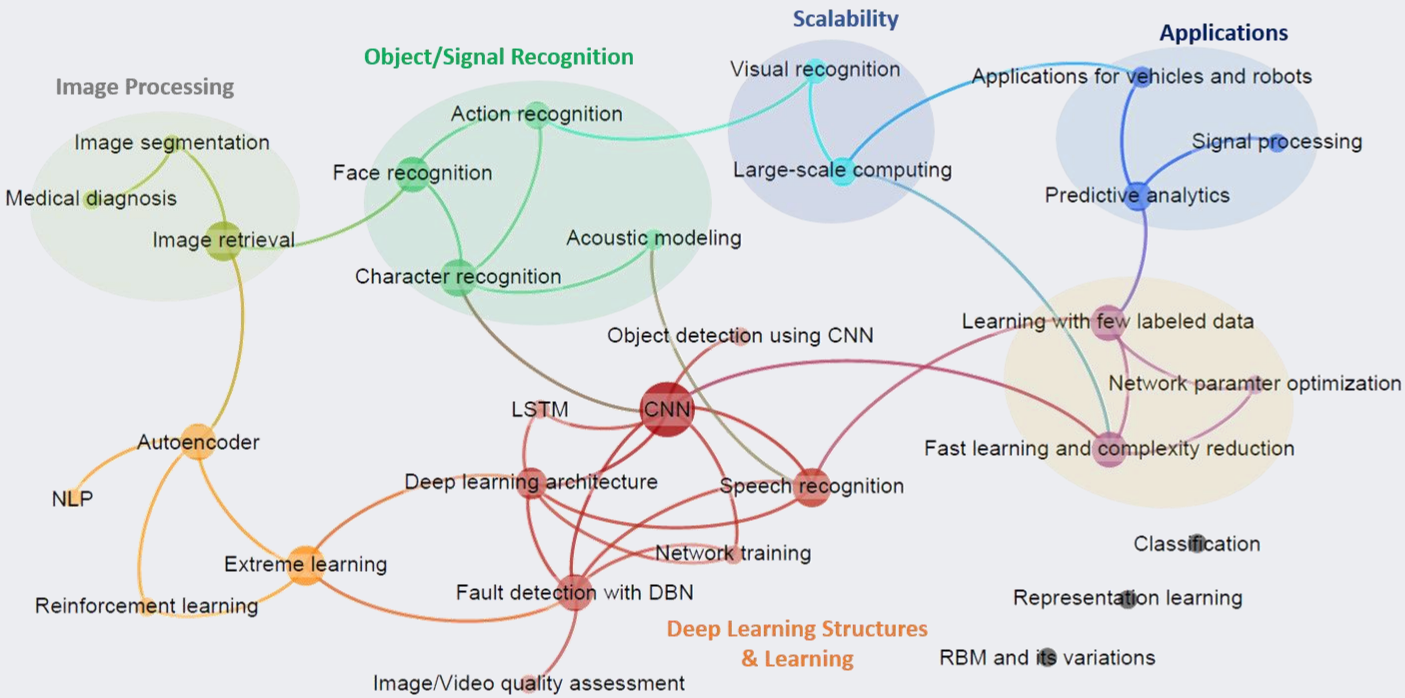


Lecture 4: Clustering

Pilsung Kang

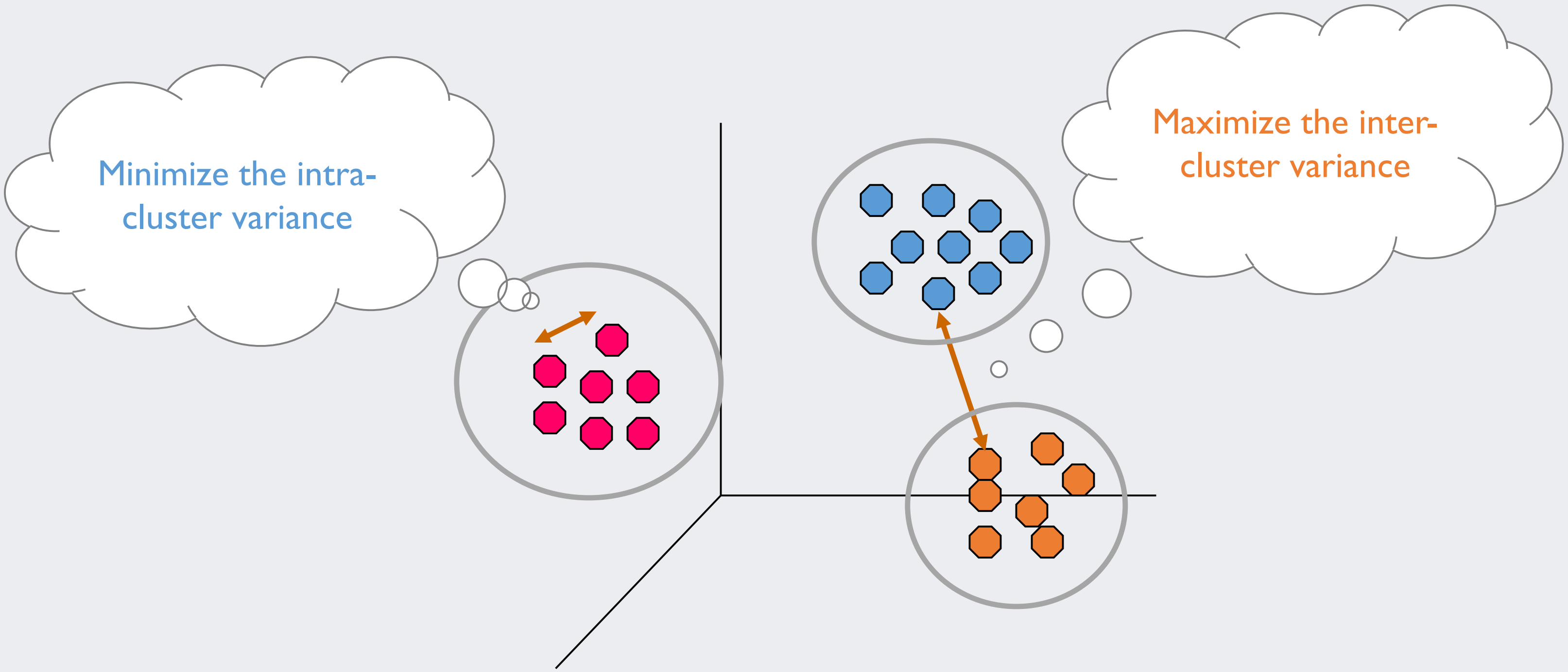
School of Industrial Management Engineering

Korea University

AGENDA

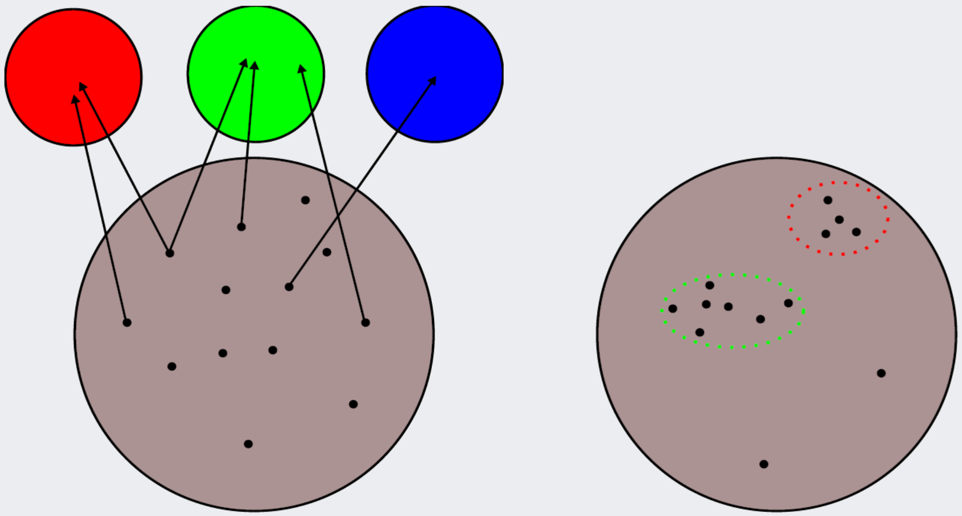
Clustering: Overview

- What is clustering?
  - ✓ Find 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



Clustering: Overview

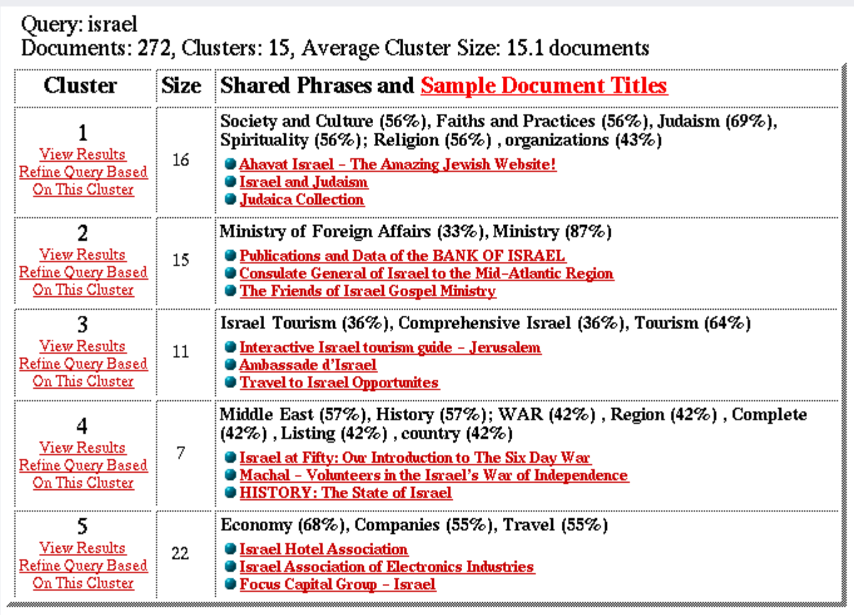
- Classification vs. Clustering
  - ✓ Classification (supervised learning)
    - The number of classes and the labels for all training instances are known
    - Goal is to find a function that links a set of input values to the target value
  - ✓ Clustering (unsupervised learning)
    - The number of clusters and memberships are unknown
    - Goal is to find an appropriate structure that can characterize the given dataset well



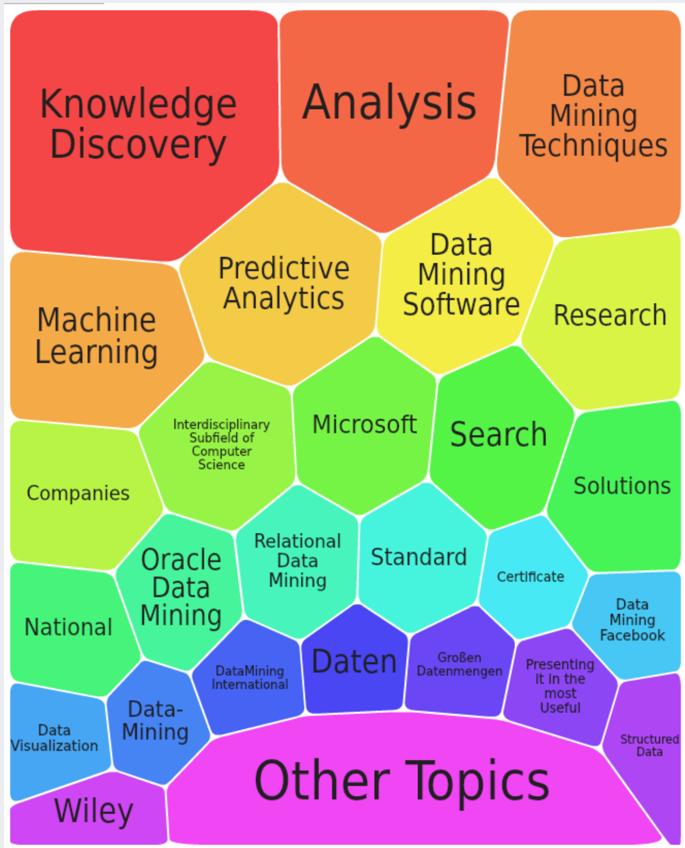
(a) Classification (b) Clustering

Clustering: Overview

- Where are clustering used?
  - ✓ "Understanding"
    - Related documents for browsing
    - Genes and proteins for similar functionalities
    - Stocks with similar price fluctuation

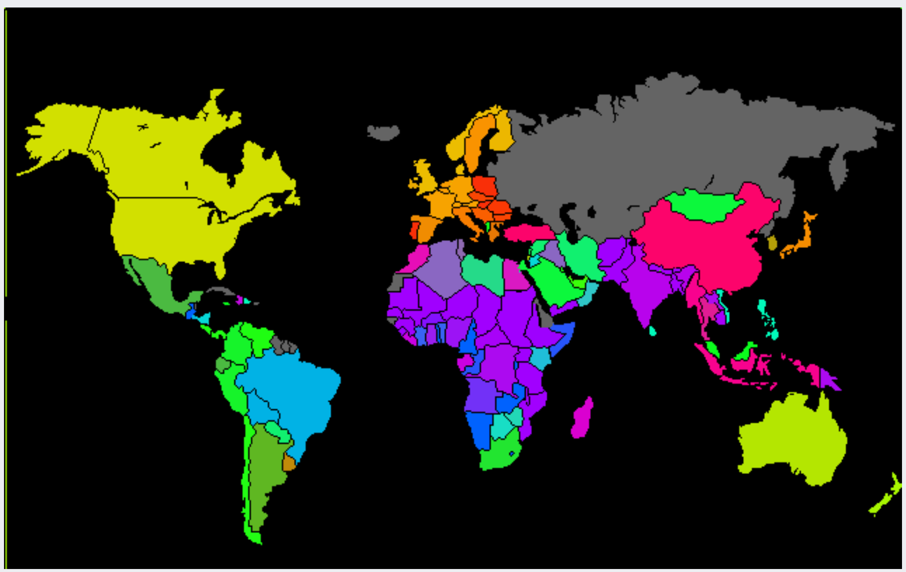

Query: israel
Documents: 272, Clusters: 15, Average Cluster Size: 15.1 documents

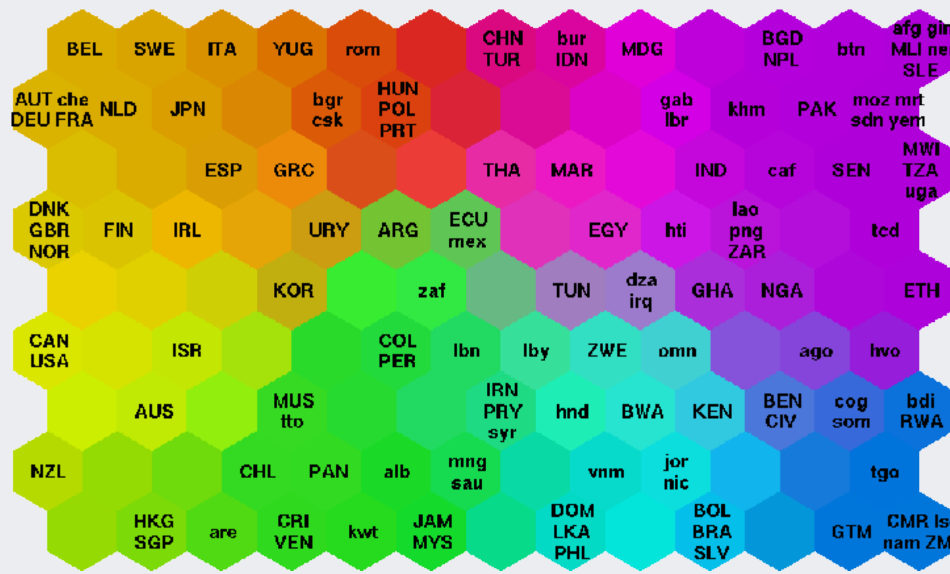
Cluster	Size	Shared Phrases and Sample Document Titles
1 View Results Refine Query Based On This Cluster	16	Society and Culture (56%), Faiths and Practices (56%), Judaism (69%), Spirituality (56%); Religion (56%) , organizations (43%) • Ahavat Israel - The Amazing Jewish Website! • Israel and Judaism • Judaica Collection
2 View Results Refine Query Based On This Cluster	15	Ministry of Foreign Affairs (33%), Ministry (87%) • Publications and Data of the BANK OF ISRAEL • Consulate General of Israel to the Mid-Atlantic Region • The Friends of Israel Gospel Ministry
3 View Results Refine Query Based On This Cluster	11	Israel Tourism (36%), Comprehensive Israel (36%), Tourism (64%) • Interactive Israel tourism guide - Jerusalem • Ambassade d'Israel • Travel to Israel Opportunites
4 View Results Refine Query Based On This Cluster	7	Middle East (57%), History (57%); WAR (42%) , Region (42%) , Complete (42%) , Listing (42%) , country (42%) • Israel at Fifty: Our Introduction to The Six Day War • Machal - Volunteers in the Israel's War of Independence • HISTORY: The State of Israel
5 View Results Refine Query Based On This Cluster	22	Economy (68%), Companies (55%), Travel (55%) • Israel Hotel Association • Israel Association of Electronics Industries • Focus Capital Group - Israel



Clustering: Overview

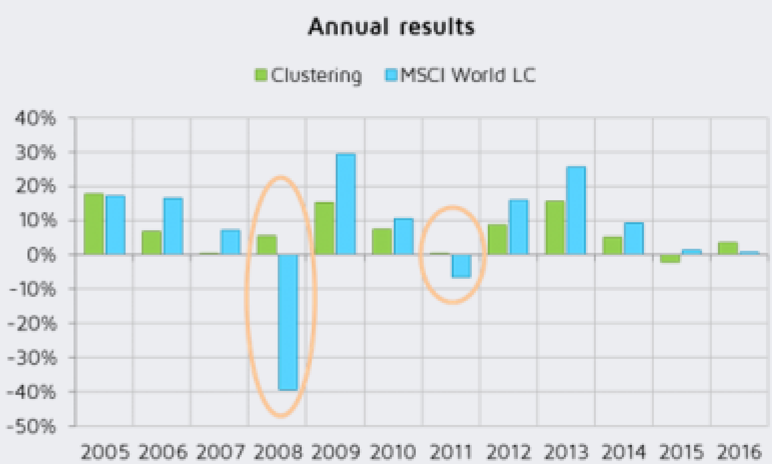
- Where are clustering used?
    - ✓ "Summarization"
        - Reduce the size of large data sets
    - ✓ Closely linked to "Visualization"





Clustering: Overview

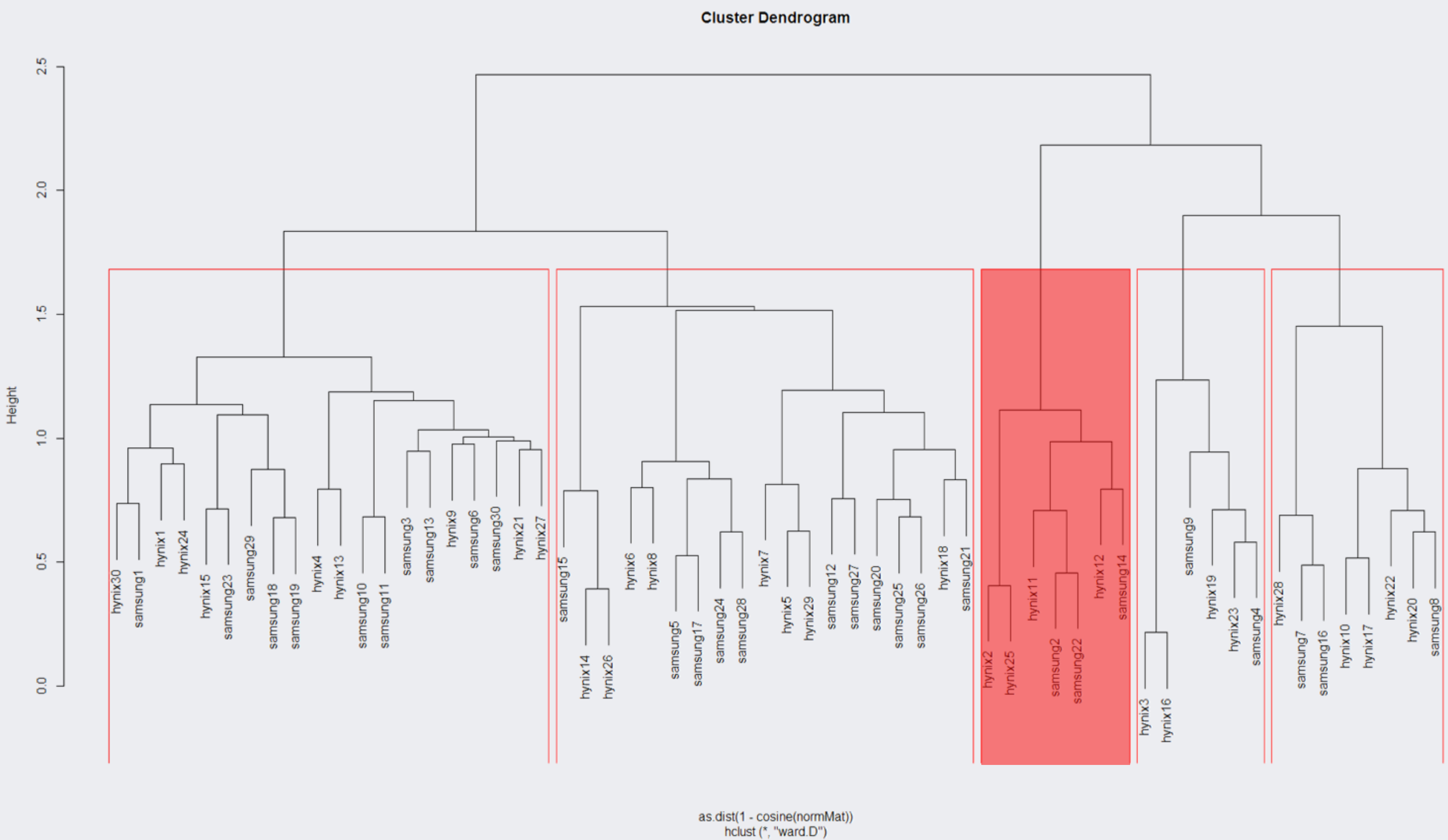
- Where are clustering used?
  - ✓ "Strategy Planning"
    - Asset management based on stock clustering
    - Stocks are clustered based on their 6 month profit and volatility
    - Select stocks from "maximum performance" and "minimum volatility group" for portfolio management



https://quantdare.com/hierarchical-clustering/

Clustering: Overview

- Where are clustering used?
  - ✓ “Strategy Planning”
    - Patent analysis to understand pros and cons compared with the rival company



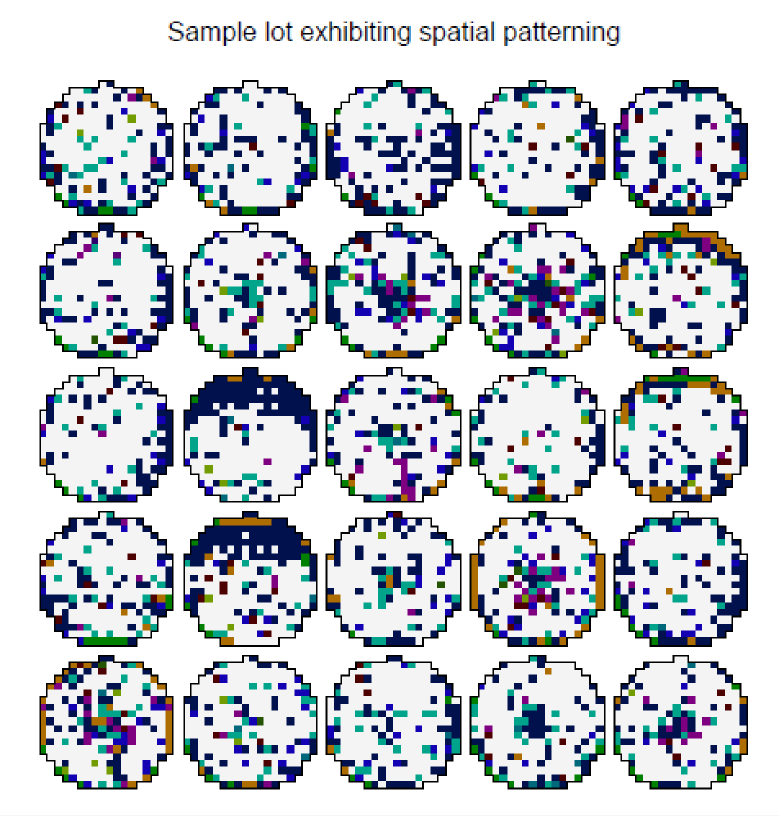
Clustering: Overview

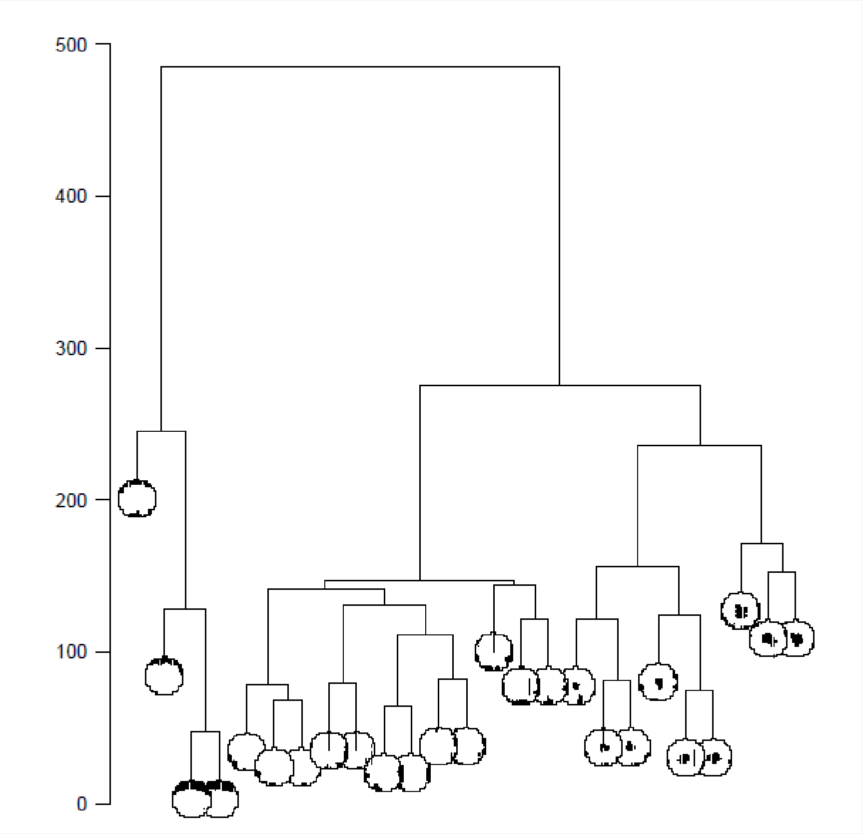
- Where are clustering used?
  - ✓ “Strategy Planning”
    - Patent analysis to understand pros and cons compared with the rival company

1	회사	일련번호	특허명	초록
2	SK하이닉스	2	멀티 레귤레이터 회로 및 이를 구비한 집적회로	본 기술에 따른 레귤레이터 회로는, 입력전압을 일정한 전압 레벨로 레귤레이팅하여 출력하도록 구성된 레귤레이터 및 복수개의 전압 생성 코드 들에 의해 결정되는 내부 저항값들에 따라 상기 레귤레이터의 출력전압을 분배한 분배전압들을 각각 출력하도록 구성된 복수개의 전압 분배회로를 포함한다.
3	SK하이닉스	11	내부 전압 생성 회로 및 그의 동작 방법	펌핑 동작을 통해 내부 전압을 생성하는 내부 전압 생성 회로에 관한 것으로, 다수의 펌핑부를 포함하며, 목표 전압 레벨에 대응하는 최종 펌핑 전압을 생성하기 위한 펌핑 전압 생성부, 및 상기 목표 전압 레벨에 대응하여 상기 다수의 펌핑부의 활성화 개수를 제어하기 위한 활성화 제어부를 구비하는 내부 전압 생성 회로가 제공된다.
4	SK하이닉스	12	자기 메모리 장치를 위한 라이트 드라이버 회로 및 자기 메모리 장치	비트라인과 소스라인 간에 접속되며, 비트라인 방향으로 인접하는 한 쌍의 자기 메모리 셀이 소스라인을 공유하는 복수의 자기 메모리 셀로 이루어진 메모리 셀 어레이를 포함하는 자기 메모리 장치를 위한 라이트 드라이버 회로로서, 정의 기록전압 공급단자와 부의 기록전압 공급단자 간에 접속되어, 라이트 인에이블 신호 및 데이터 신호에 따라 정의 기록전압 또는 부의 기록전압에 의한 전류를 비트라인에 선택적으로 공급하는 스위칭부를 포함하는 자기 메모리 장치를 제공한다.
5	SK하이닉스	25	전압 레귤레이터 및 전압 레귤레이팅 방법	전압 레귤레이터는 출력전압을 전압 출력단으로 출력하는 전압 출력부와, 제1 제어코드의 제어에 따라 분배저항값을 조절하는 제1 저항분배 스테이지와, 제1 저항분배 스테이지에서 결정된 분배 저항값을 제2 제어코드의 제어에 따라 조절하는 제2 저항분배 스테이지;를 포함하며, 전압 출력단을 통해서 출력되는 출력전압의 전압레벨은 제1 및 제2 저항분배 스테이지를 통해서 결정된 상기 분배 저항값과, 기준저항의 저항값 비율에 따라 조절되는 것을 특징으로 한다.
6	삼성전자	2	전압 공급 장치 및 그것을 포함한 불휘발성 메모리 장치	본 발명에 따른 전압 공급 장치는 전원 전압을 승압하고, 상기 승압된 전압을 출력 라인으로 제공하기 위한 전하 펌프 및 상기 출력 라인의 전압 레벨을 목표 전압 레벨로 유지하기 위한 전압 제어 회로를 포함한다. 본 발명에 따른 상기 전압 제어 회로는 웰 상에 형성된 제 1 영역 및 제 2 영역을 포함하고, 상기 제 1 영역 및 제 2 영역 사이의 리치 스루(reach through)를 이용하여 상기 출력 라인의 전압 레벨을 제어하기 위한 리치 스루 소자를 포함한다.
7	삼성전자	14	파워 공급 회로 및 이를 구비하는 상 변화 메모리 장치	파워 공급 회로 및 이를 구비하는 상 변화 메모리 장치가 개시된다. 본 발명의 제 1 실시예에 따른 반도체 메모리 장치는 파워 공급 회로, 스위치들 및 선택기들을 구비한다. 파워 공급 회로는 상기 블록들의 메모리 셀들에 사용되는 제 1 전압 및 제 2 전압을 생성한다. 스위치들은 상기 파워 공급 회로와 상기 제 1 전압이 전달되는 제 1 라인 및 상기 제 2 전압이 전달되는 제 2 라인으로 연결되고, 제어 신호에 응답하여 상기 제 1 전압 및 제 2 전압 중 하나를 대응되는 블록으로 인가한다. 선택기들은 블록 선택 신호 및 디스차아지 성공신호에 응답하여, 상기 제어 신호를 생성한다. 본 발명에 따른 파워 공급 회로 및 이를 구비하는 상 변화 메모리 장치는 셀 블록마다 별도의 파워 스위치를 구비함으로써 파워 공급 회로의 동작 시간 및 동작 전류를 감소시킬 수 있다. 또한, 기입 전압을 디스차아지한 후 다른 레벨의 전압을 공급함으로써, 상 변화 메모리 장치의 오작동이 방지될 수 있다.
8	삼성전자	22	전압 안정화 장치 및 그것을 포함하는 반도체 장치 및 전압 생성 방법	본 발명은 전압 안정화 장치 및 그것을 이용하는 반도체 장치에 관한 것이다. 본 발명의 기술적 사상의 실시예에 따른 전압 안정화 장치는 제 1 전압을 생성하는 제 1 레귤레이터 및 상기 제 1 전압보다 낮은 제 2 전압을 생성하는 제 2 레귤레이터를 포함하되, 상기 제 2 레귤레이터는 상기 제 1 전압의 레벨과 미리 정해진 기준 전압의 레벨의 비교 결과에 기초하여 상기 제 1 전압 또는 상기 제 1 전압보다 높은 제 3 전압을 선택적으로 이용하여 상기 제 2 전압을 생성한다. 본 발명의 기술적 사상의 실시 예에 따르면 제 1의 전압≥제 2의 전압의 관계를 유지하면서, 동시에 제 2의 전압을 고속으로 전위 변환 시킬 수 있다.

Clustering: Overview

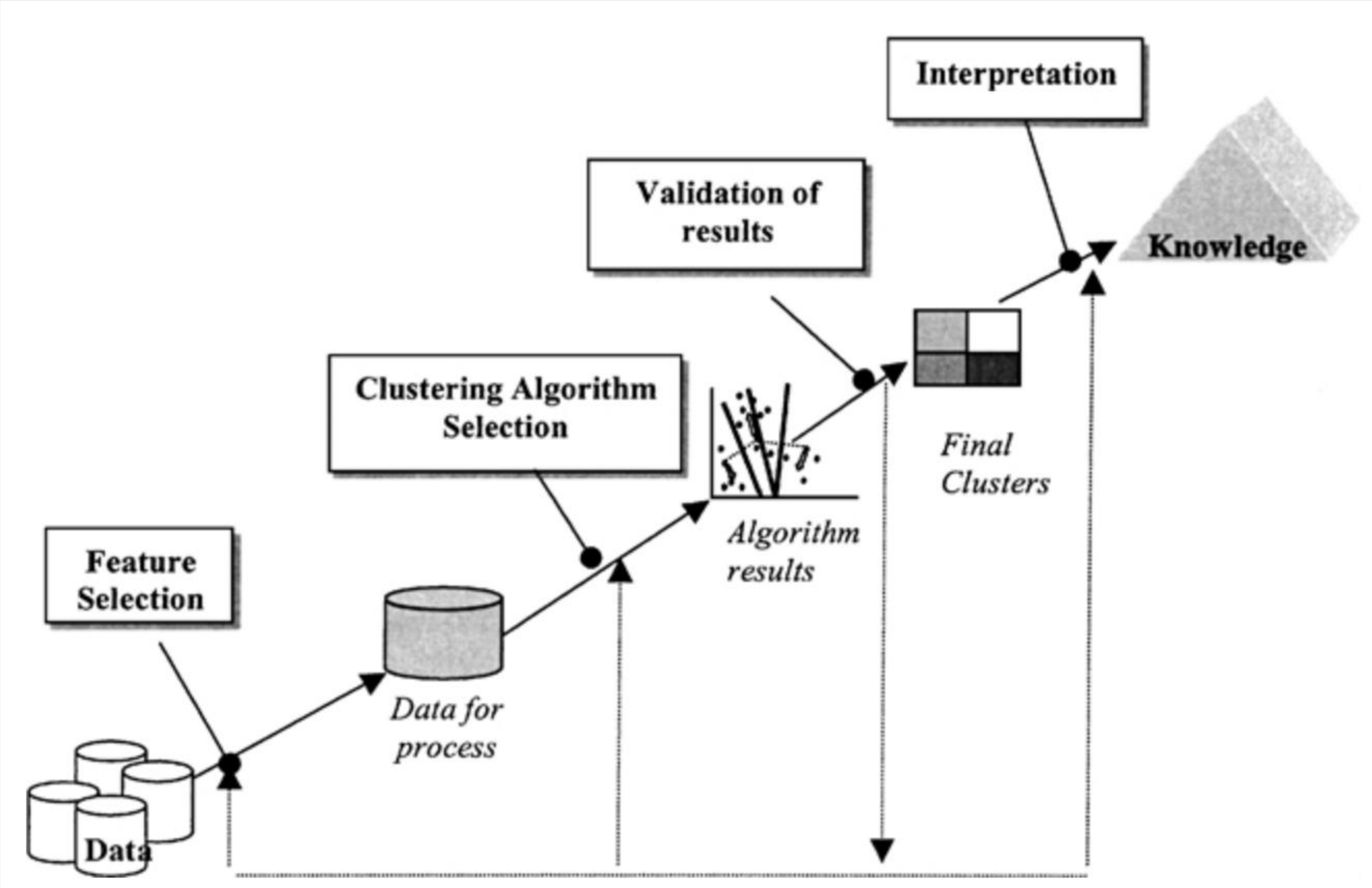
- Where are clustering used?
  - ✓ In-depth analysis





Clustering: Overview

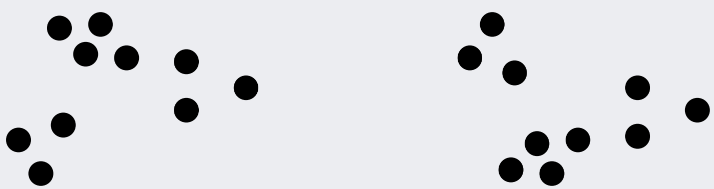
- Standard clustering procedure



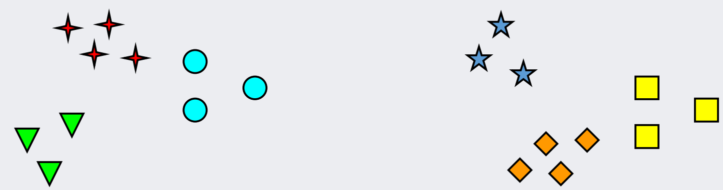
http://web.itu.edu.tr/sgunduz/courses/verimaden/paper/validity_survey.pdf

Clustering: Issues

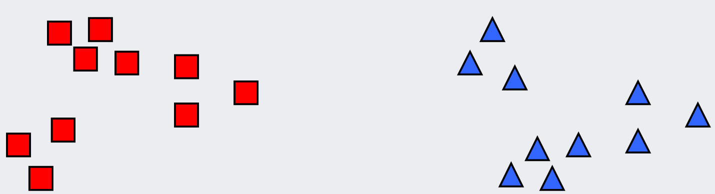
- How many clusters are optimal?



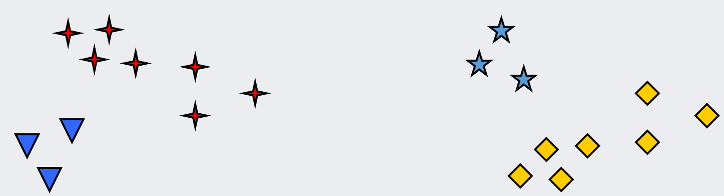
How many clusters?



Six Clusters



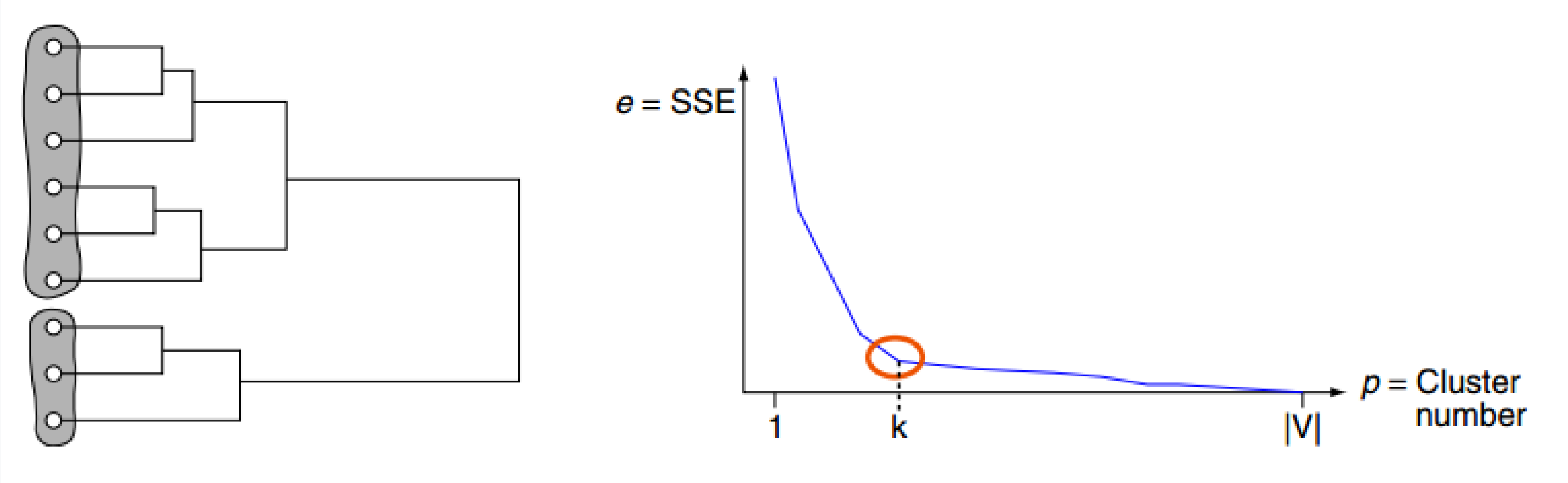
Two Clusters



Four Clusters

Clustering: Issues

- How many clusters are optimal?
  - ✓ Use a clustering validity measure to evaluate the clustering result
  - ✓ Find the elbow point



Clustering: Issues

- How to evaluate the clustering result?
  - ✓ There is no globally accepted validity measure
  - ✓ Because clustering is an unsupervised learning task, we do not know the exact answer
- Three categories for clustering validity measures
  - ✓ External: Compare the clustering structure with the known answer (unrealistic)
  - ✓ Internal: Focusing on the compactness of clusters
  - ✓ Relative: Focusing on both the compactness of clusters and separation between clusters

Clustering: Issues

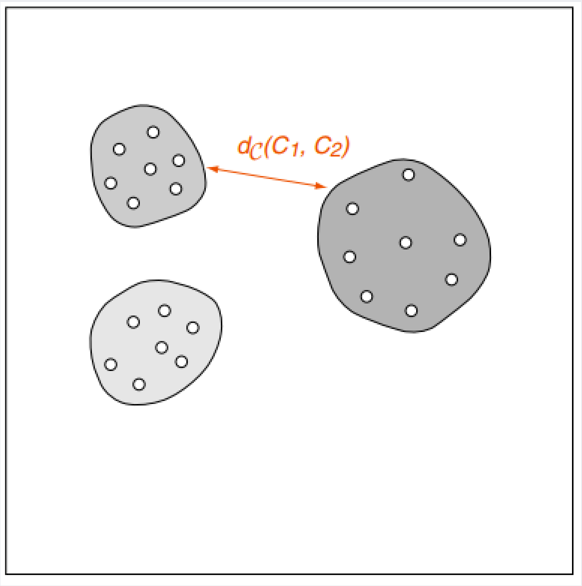
- Examples of clustering validity measures

External	Internal	Relative
Rand Statistic	Cophenetic Correlation Coefficient	Dunn family of indices
Jaccard Coefficient	Sum of Squared error (SSE)	Davies-Bouldin (DB) index
Folks and Mallows index	Cohesion and separation	Semi-partial R-squared
(Normalized) Hurbert Γ statistic		SD validity index
		Silhouette

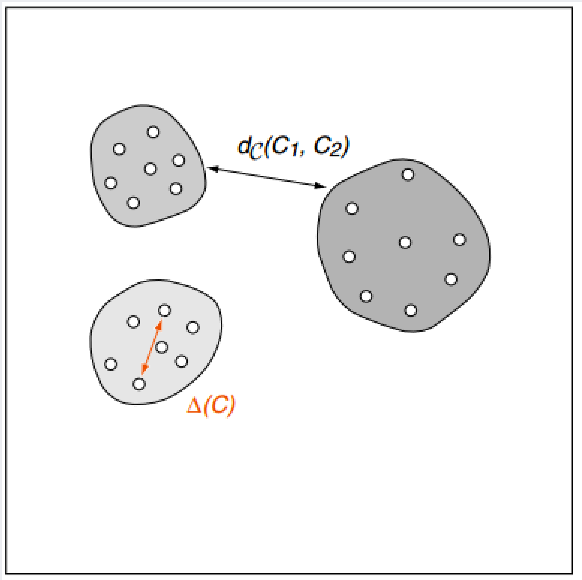
Clustering: Issues

- Clustering Validity Measure Example: <u>Dunn Index</u>
  - ✓ If the clustering is well performed,
    - The value of (1) will be large and the values of (2) and (3) will be small

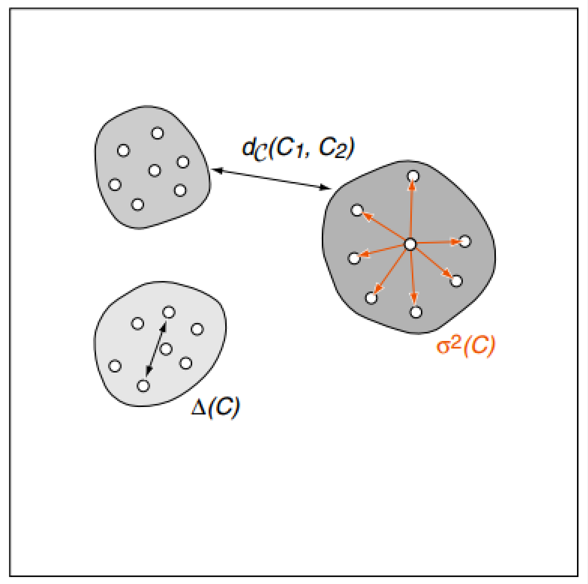
(1) Distance between two clusters



(2) Diameter of a cluster

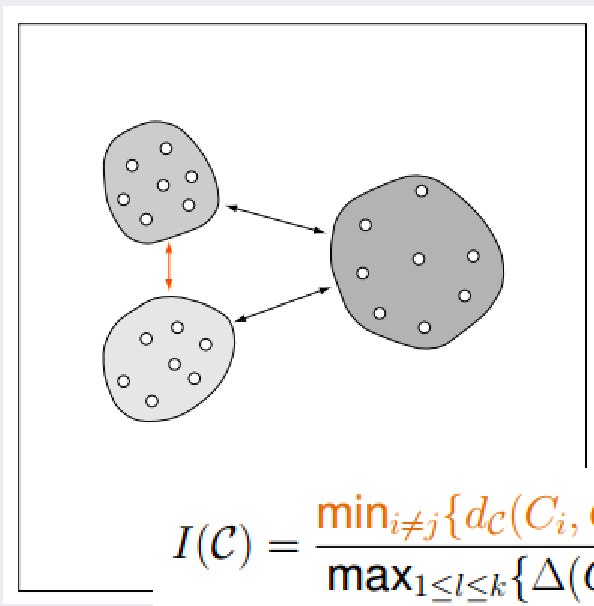


(3) Scatter within a cluster (SSE)



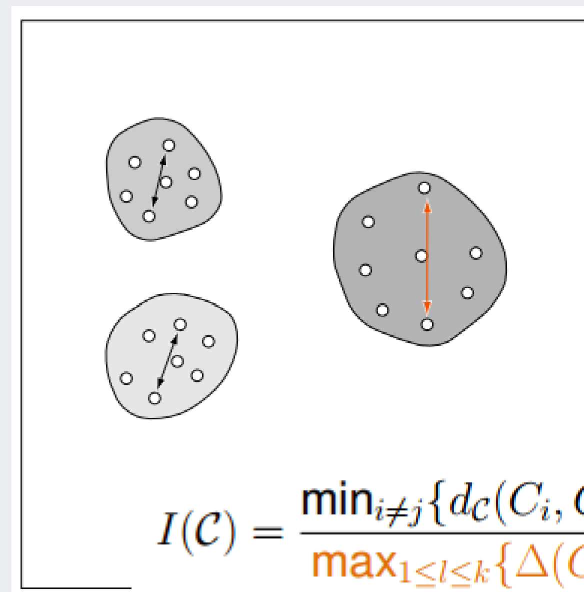
Clustering: Issues

- Clustering Validity Measure Example: Dunn Index
  - ✓ Dunn index is defined the ratio of (1) the minimum distance between two clusters to (2) the maximum diameter of the clusters



$$I(\mathcal{C}) = \frac{\min_{i \neq j}\{d_{\mathcal{C}}(C_i, C_j)\}}{\max_{1 \leq l \leq k}\{\Delta(C_l)\}},$$

$$I(\mathcal{C}) \rightarrow \max$$



$$I(\mathcal{C}) = \frac{\min_{i \neq j}\{d_{\mathcal{C}}(C_i, C_j)\}}{\max_{1 \leq l \leq k}\{\Delta(C_l)\}},$$

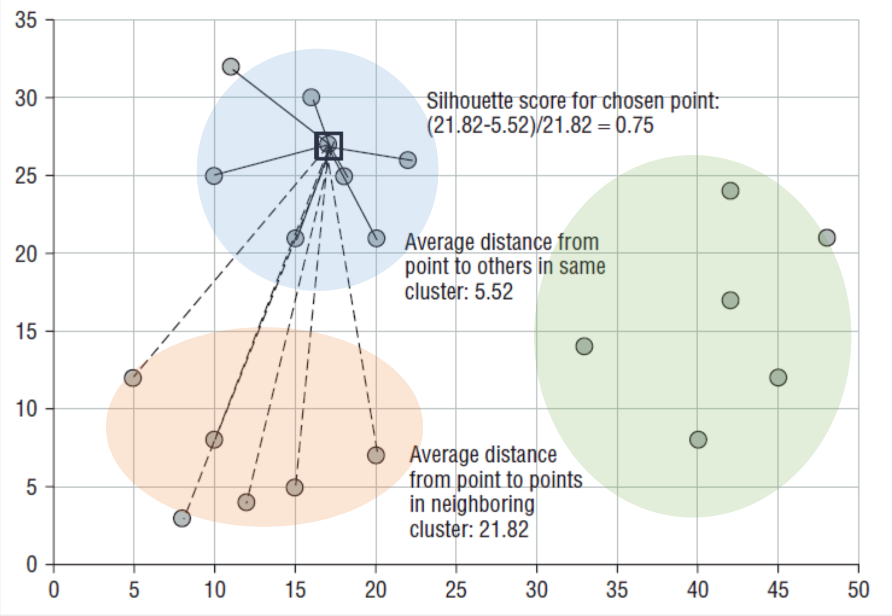
$$I(\mathcal{C}) \rightarrow \max$$

Clustering: Issues

- Clustering Validity Measure Example: <u>Silhouette</u>
  - ✓ a(i): the average distance between an instance i and the other instances in the same cluster
  - ✓ b(i) the minimum of the average distances between an instance i and the instances is a cluster to which the instance i does not belong

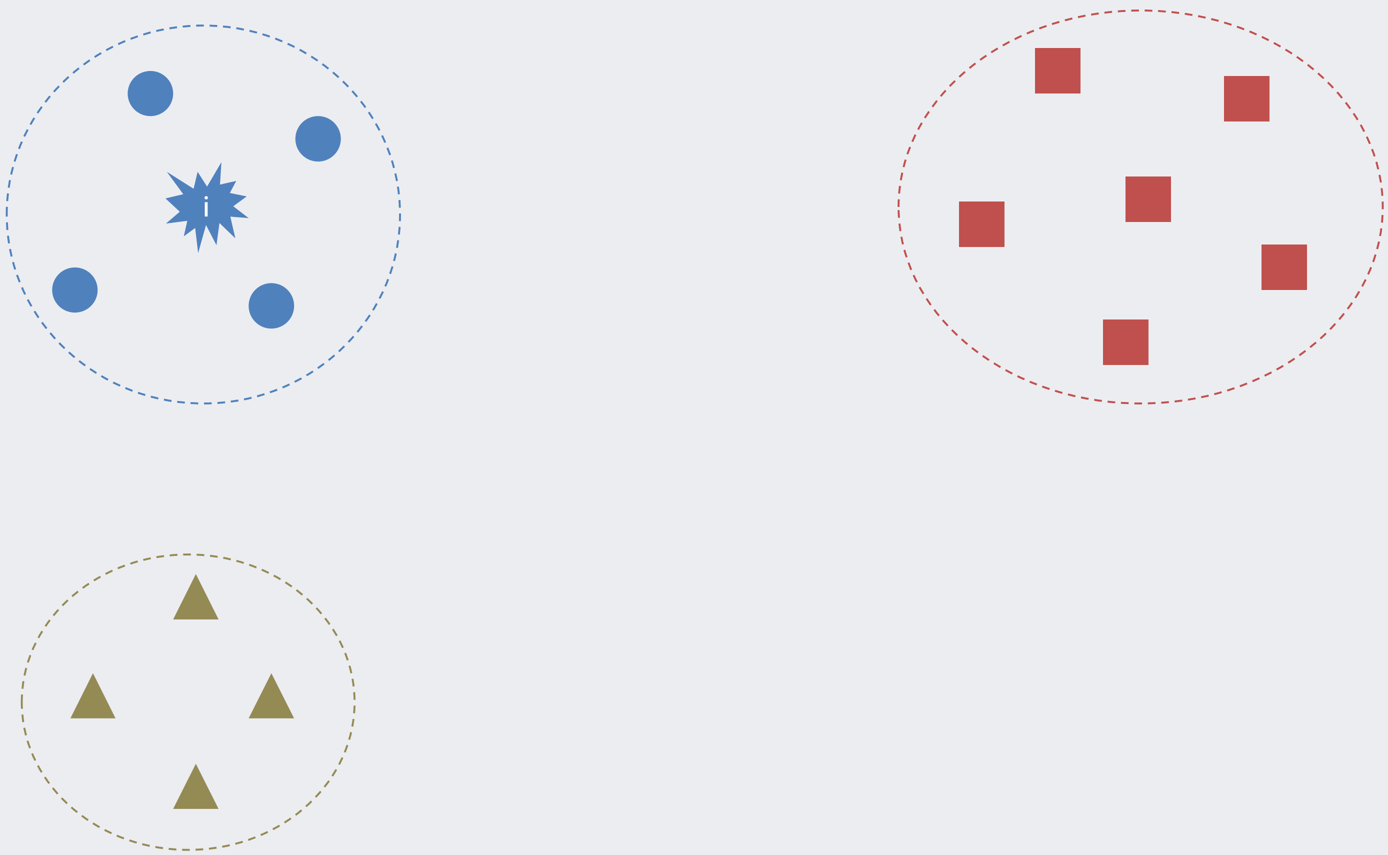
$$s(i) = \frac{b(i) - a(i)}{max\{a(i), b(i)\}}$$

$$s(i) = \begin{cases} 1 - a(i)/b(i), & if \ a(i) < b(i) \\ 0, & if \ a(i) = b(i) \\ b(i)/a(i) - 1, & if \ a(i) > b(i) \end{cases}$$



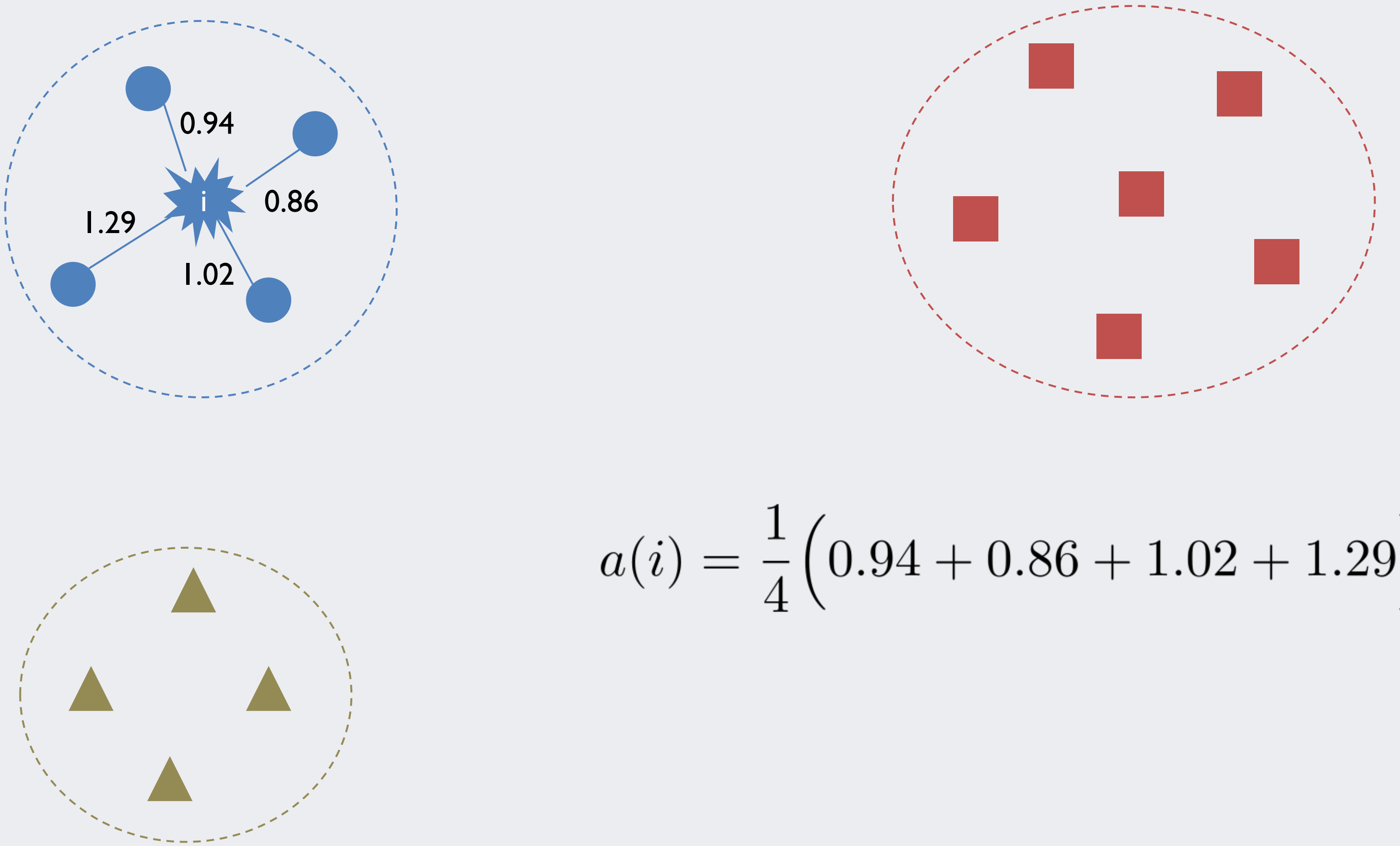
Clustering: Issues

- Clustering Validity Measure Example: Silhouette



Clustering: Issues

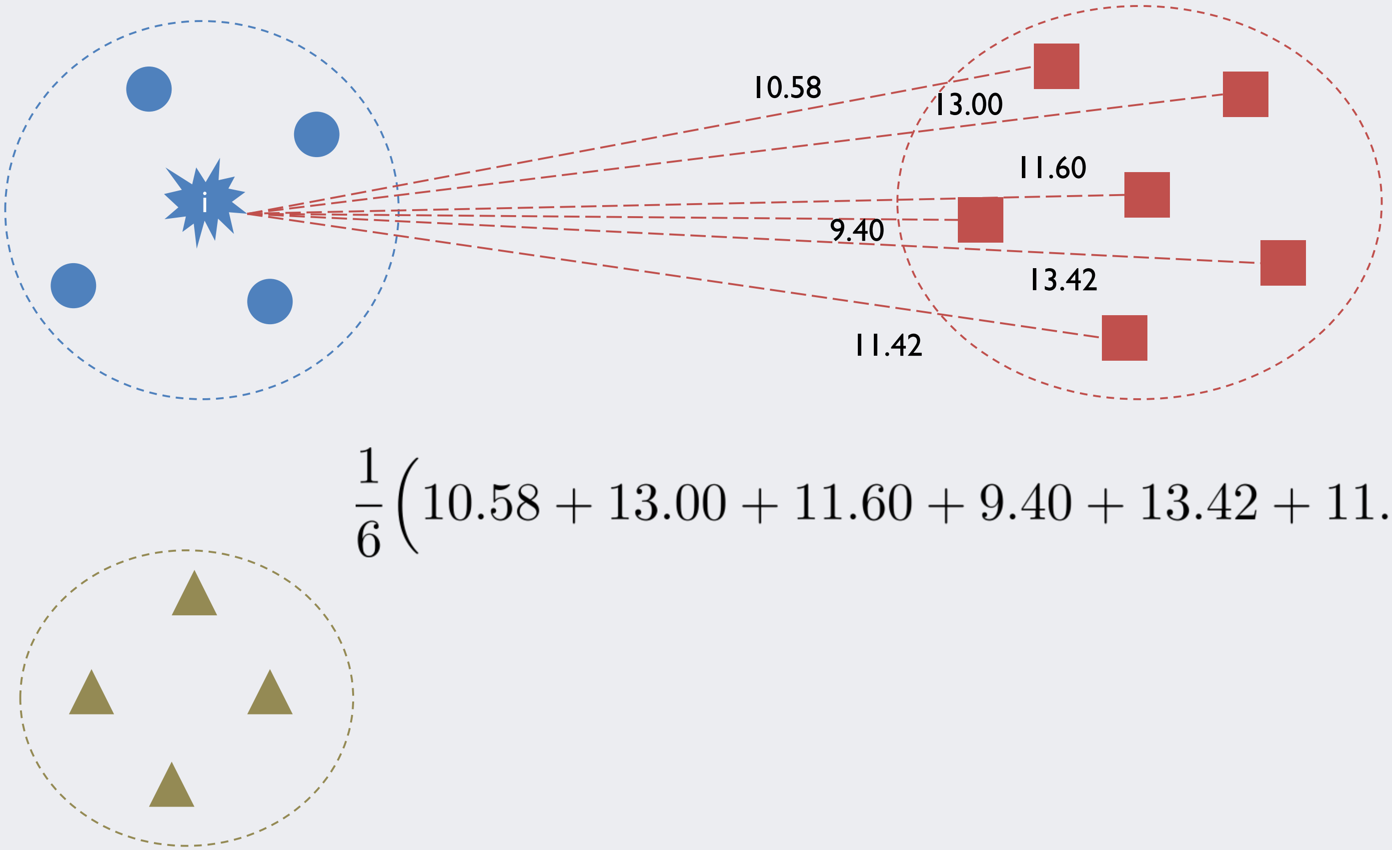
- Clustering Validity Measure Example: Silhouette



$$a(i) = \frac{1}{4}\Big(0.94 + 0.86 + 1.02 + 1.29\Big) = 1.03$$

Clustering: Issues

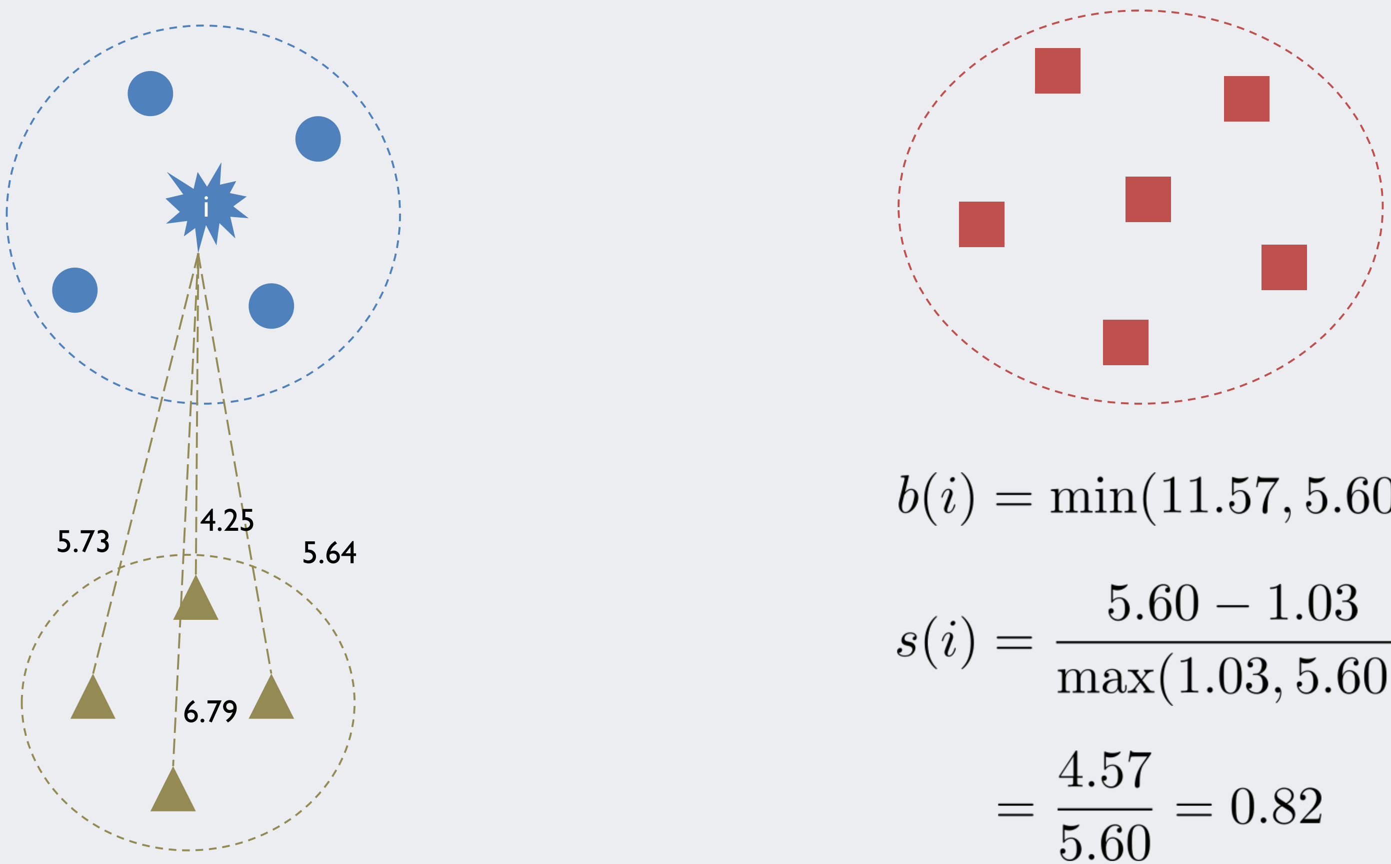
- Clustering Validity Measure Example: Silhouette



$$\frac{1}{6}\Big(10.58 + 13.00 + 11.60 + 9.40 + 13.42 + 11.42\Big) = 11.57$$

Clustering: Issues

- Clustering Validity Measure Example: Silhouette



$$\frac{1}{4}\Big(5.73 + 6.79 + 4.25 + 5.64\Big) = 5.60$$

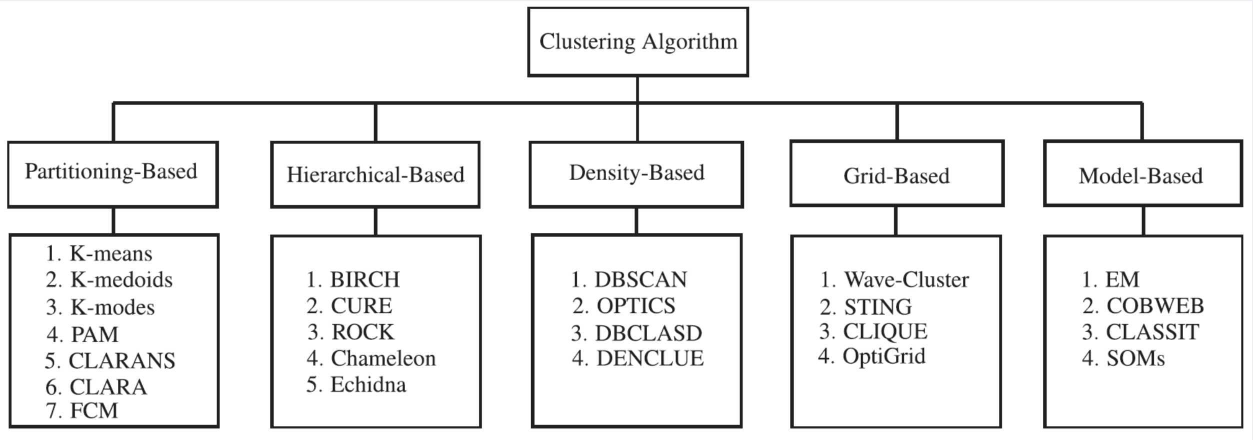
$$b(i) = \min(11.57, 5.60) = 5.60$$

$$s(i) = \frac{5.60 - 1.03}{\max(1.03, 5.60)}$$

$$= \frac{4.57}{5.60} = 0.82$$

Clustering: Types

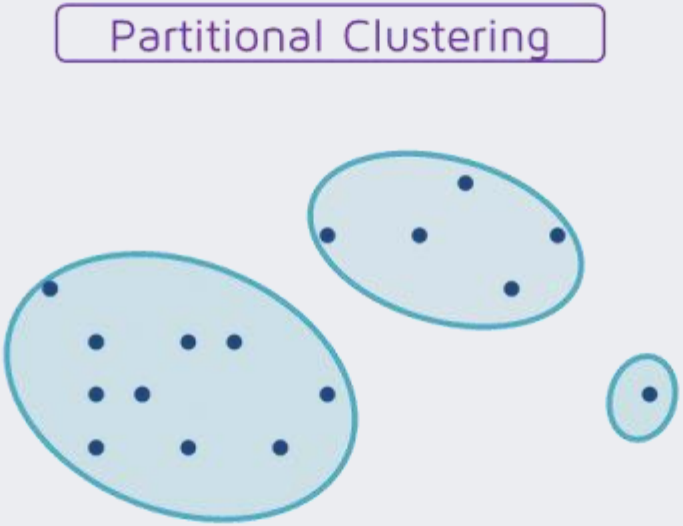
- Hard clustering vs. Soft clustering
  - ✓ Hard Clustering (Crisp Clustering)
    - Results in non-overlapping clusters
    - Each instance belongs to only one cluster



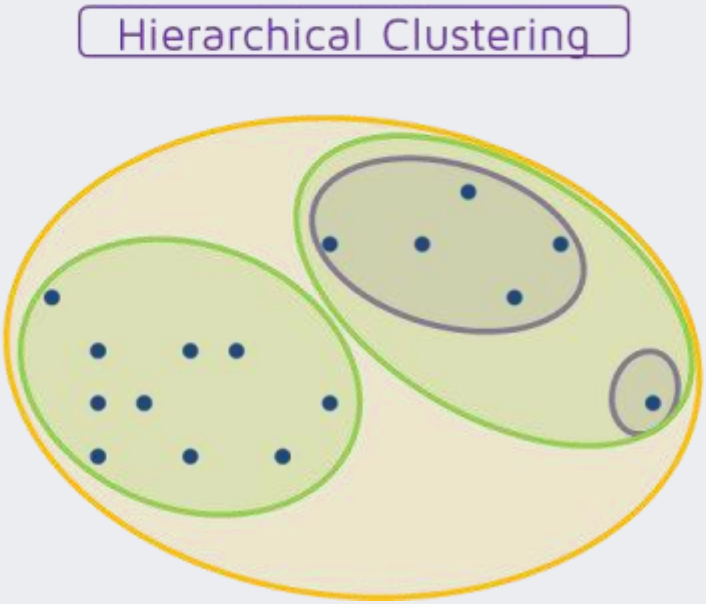
  - ✓ Soft Clustering (Fuzzy Clustering)
    - Possible to result in overlapping clusters
    - Each instance can belong to more than two clusters

Clustering: Algorithms

- Partitional clustering
  - ✓ Divide data into non-overlapping subsets such that each data object is in exactly one subset



- Hierarchical clustering
  - ✓ A set of nested clusters organized as a hierarchical tree



AGENDA

K-Means Clustering

- K-Means Clustering (KMC)
  - ✓ Partitional clustering approach
    - Each cluster is associated with a centroid
    - Each point is assigned to the cluster with the closest centroid
    - Number of cluster, K, must be specified

$$\mathbf{X} = C_1 \cup C_2 \ldots \cup C_K, \quad C_i \cap C_j = \phi, \quad i \neq j$$

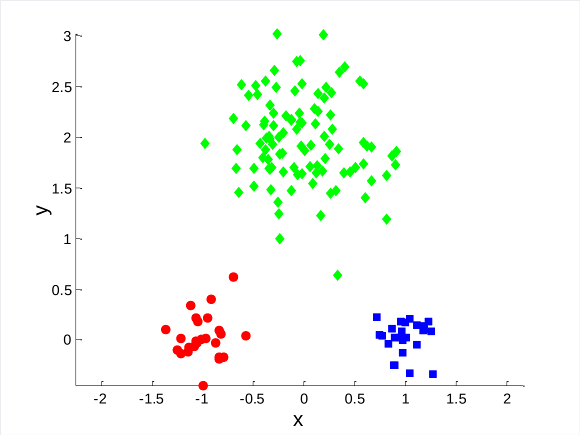
$$\arg\min_{\mathbf{C}} \quad \sum_{i=1}^{K} \sum_{\mathbf{x}_j \in C_i} ||\mathbf{x}_j - \mathbf{c}_i||^2$$

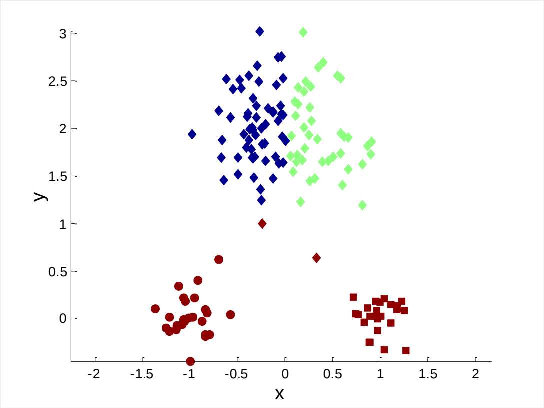
K-Means Clustering

- K-Means Clustering Procedure

```
1: Select K points as the initial centroids.
2: repeat
3:     Form K clusters by assigning all points to the closest centroid.
4:     Recompute the centroid of each cluster.
5: until The centroids don't change
```

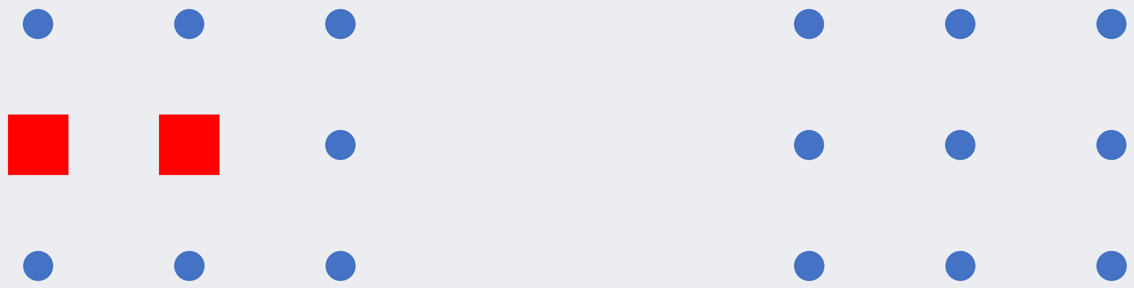
✓ Initial centroids are often chosen randomly: clustering results vary according to the initial centroid selection



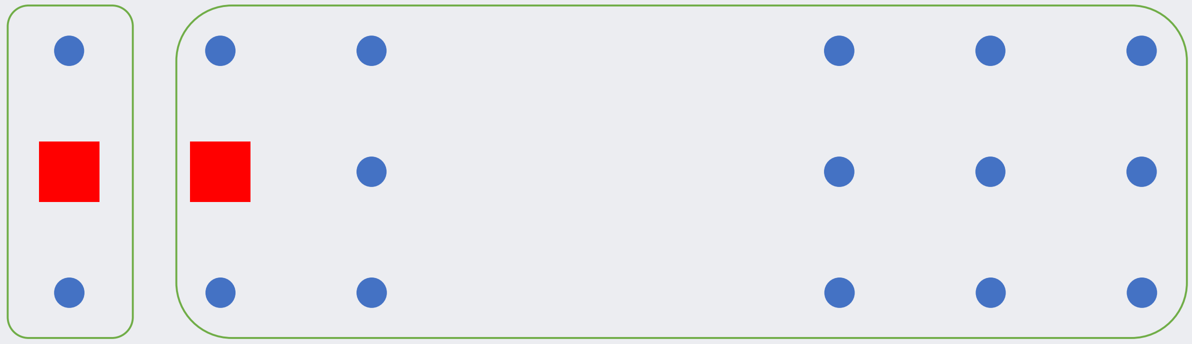


K-Means Clustering

- Example
  - ✓ Step 1: Initializing K centroids

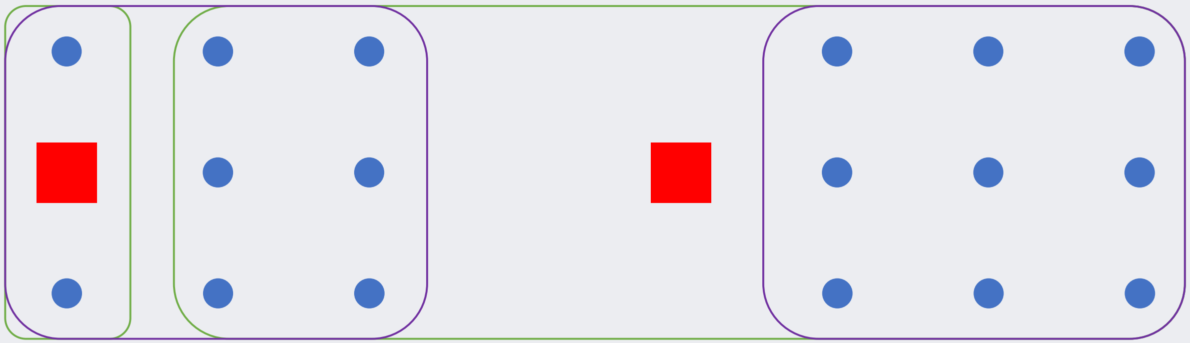


  - ✓ Step 2-1 (1st): Assign each instance to the closest center
  - ✓ Step 2-2 (1st): Re-compute the centroids based on the assigned instances

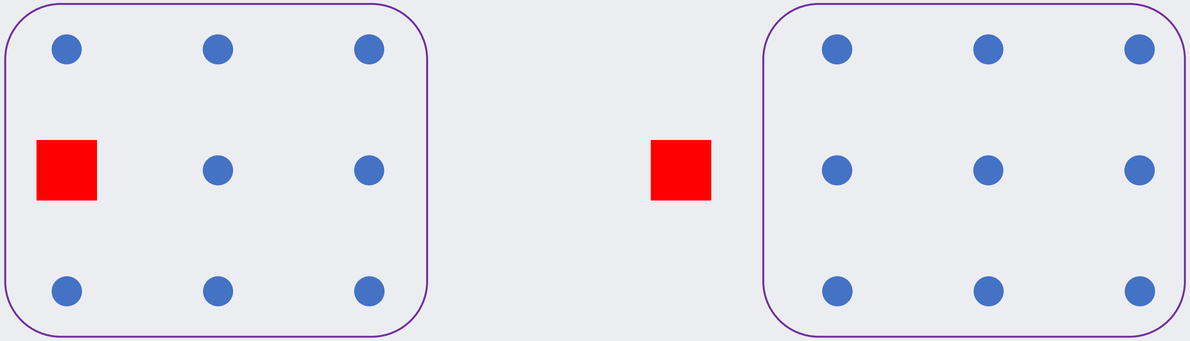


K-Means Clustering

- Example
  - ✓ Step 2-1 (2nd): Assign each instance to the closest center



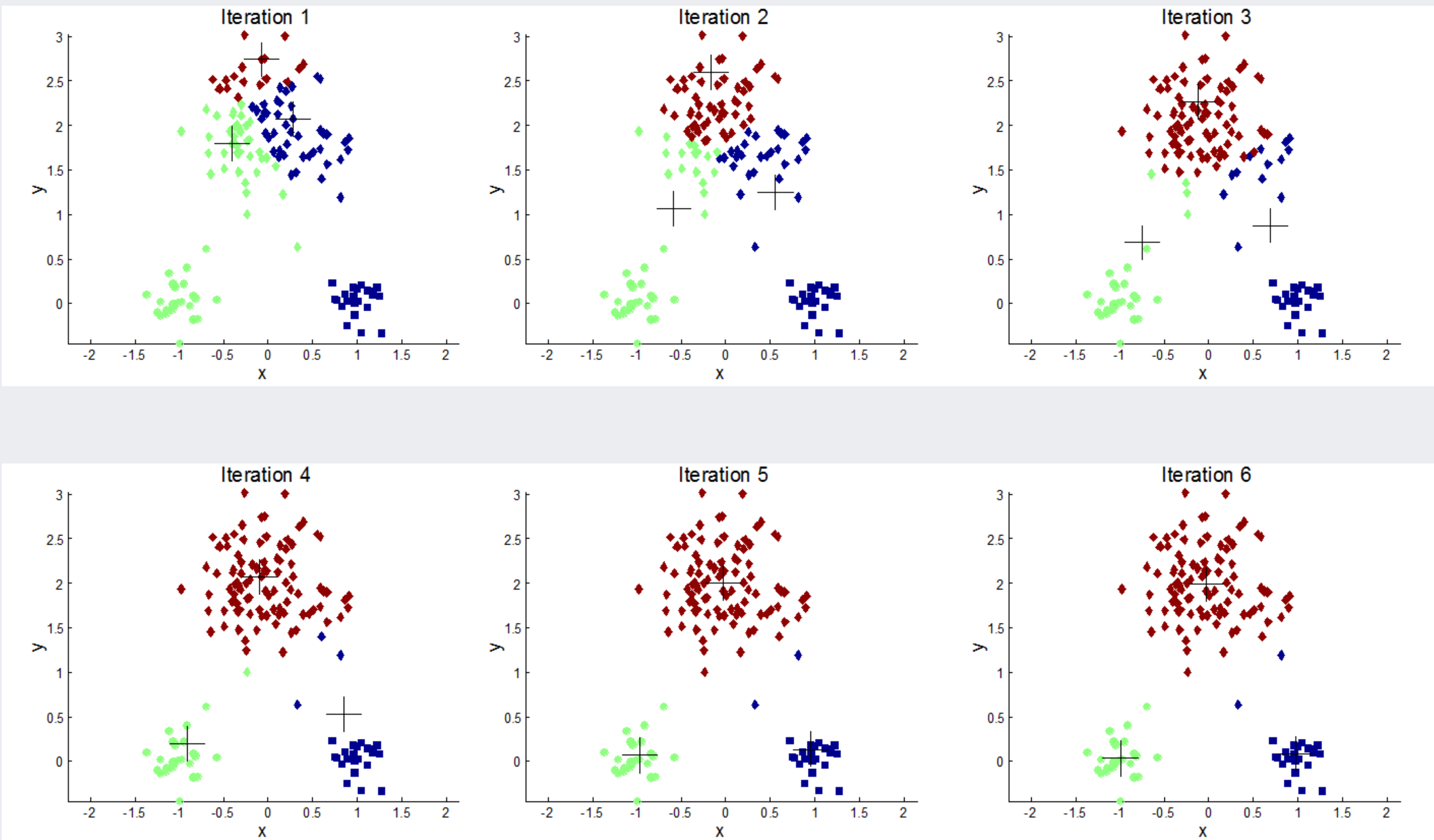
  - ✓ Step 2-2 (2nd): Re-compute the centroids based on the assigned instances



  - ✓ Stop the algorithm because there is no change for centroids and membership assignment

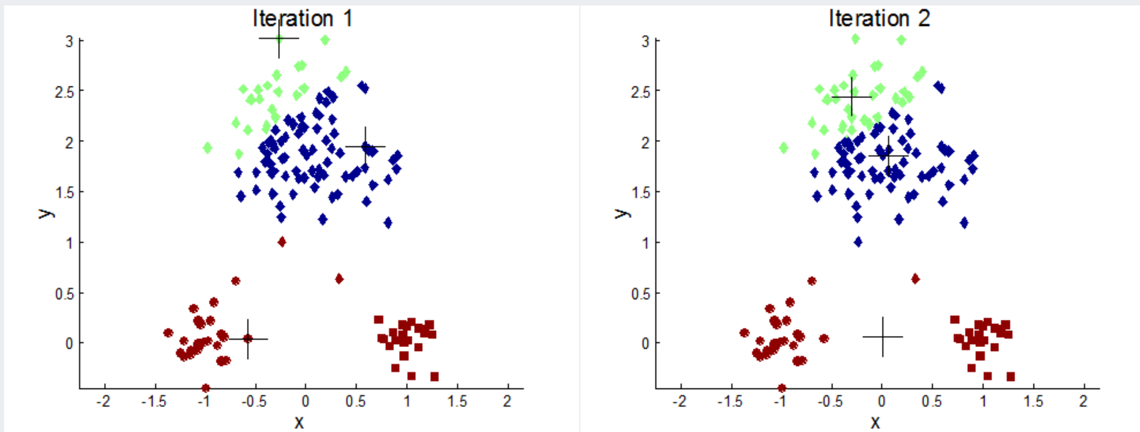
K-Means Clustering

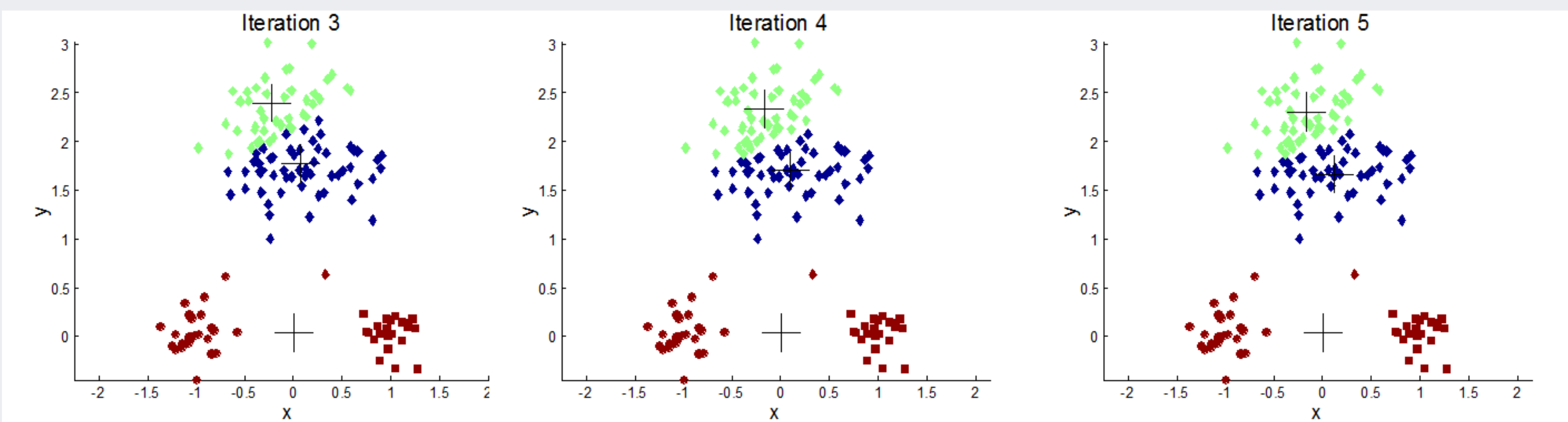
- Effect of initial centroids
  - ✓ Desirable centroid selection



K-Means Clustering

- Effects of initial centroids
  - ✓ Undesirable centroid selection

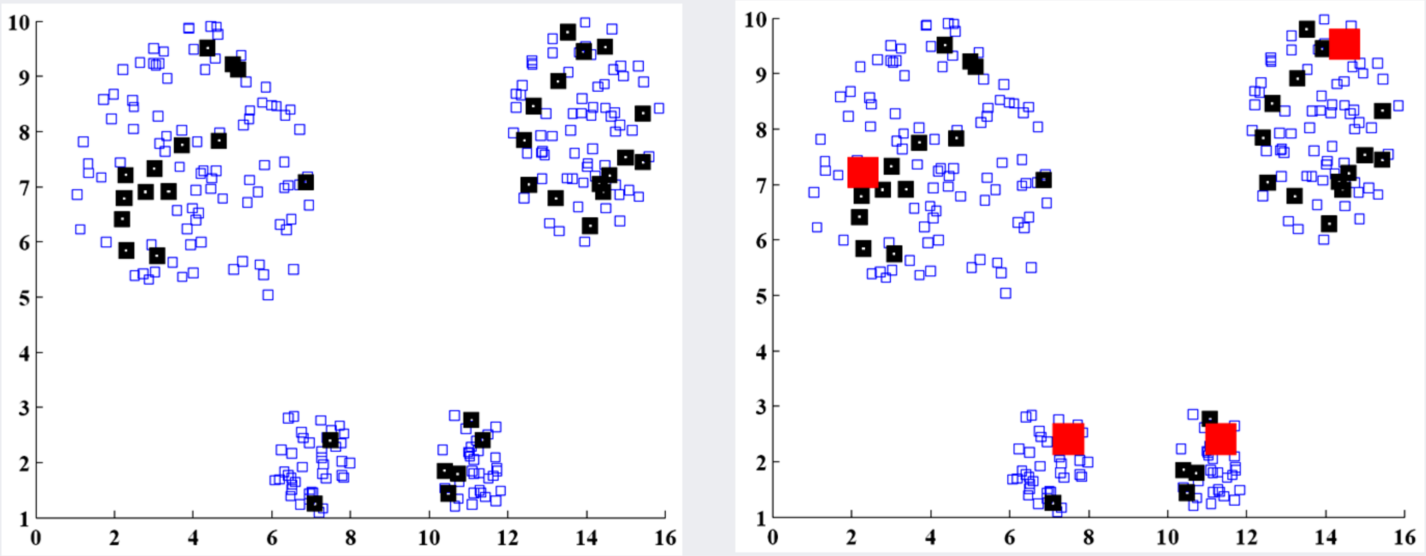




K-Means Clustering

- Some remedies for initial centroid selection
  - ✓ Multiple runs
  - ✓ Sample and use hierarchical clustering to determine initial centroids
  - ✓ Preprocessing & Postprocessing

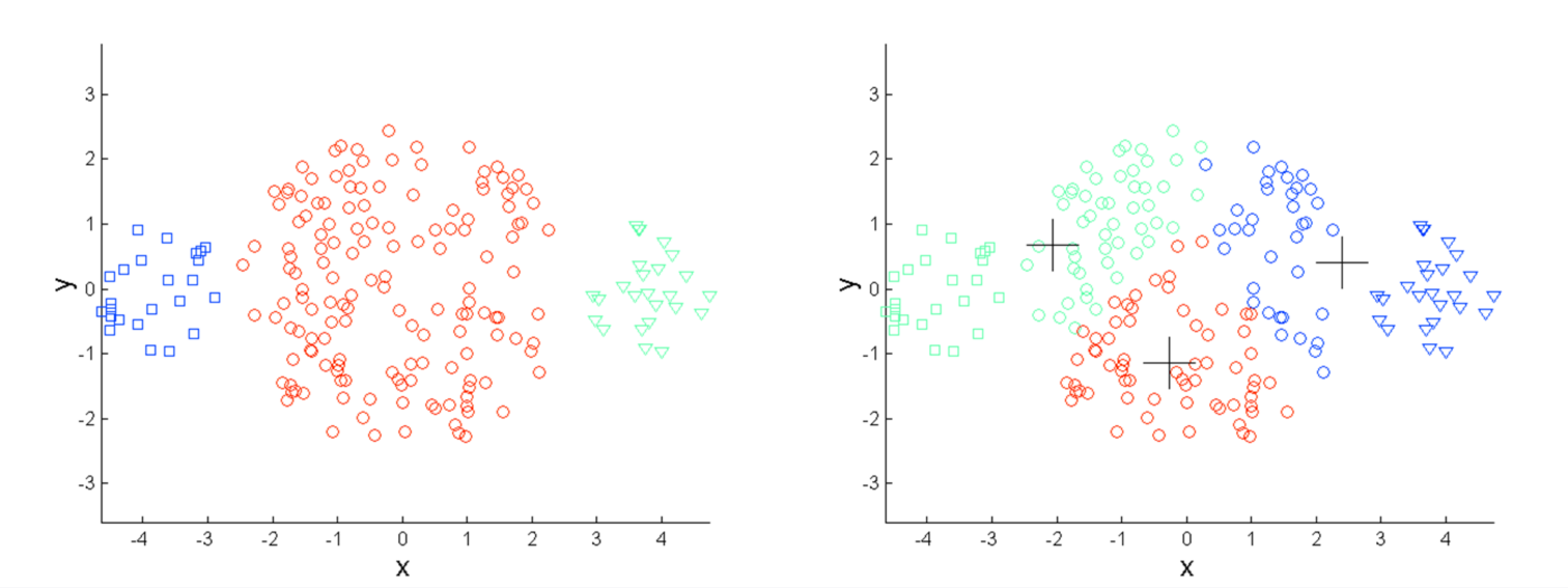
$$\mathcal{L}(\mathbf{x}_s|\mathbf{S},\mathbf{C}) = d_G(\mathbf{x}_s,\mathbf{S}) \times \frac{1}{1+\exp\big(-d_R(\mathbf{x}_s,\mathbf{S})\big)}$$



Pilsung Kang and Sungzoon Cho. (2009). K-Means clustering seeds initialization based on centrality, sparsity, and isotropy. *The 13th International Conference on Intelligent Data Engineering and Automated Learning (IDEAL 2009)*, Burgos, Spain. E. Corchado and H. Yin (Eds.), ***Lecture Notes in Computer Science LNCS 5788***, 109-117.

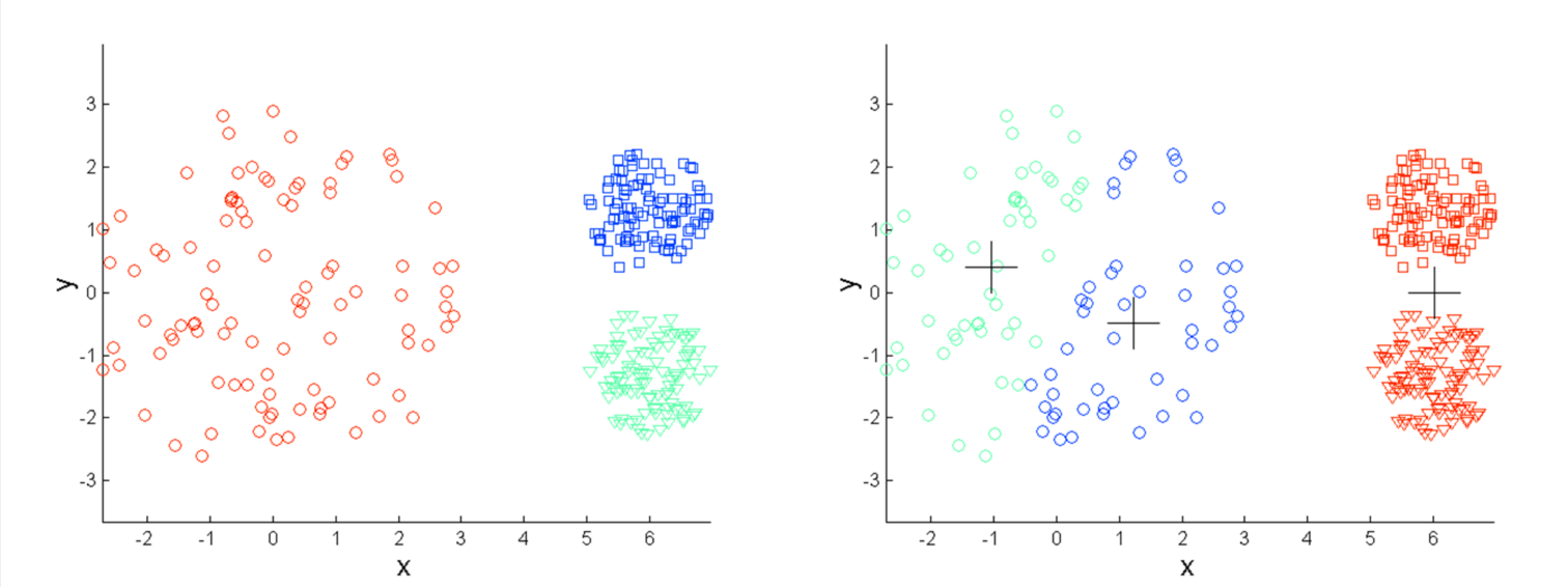
K-Means Clustering

- Limitations of K-Means Clustering
  - ✓ Cannot cope with different sizes



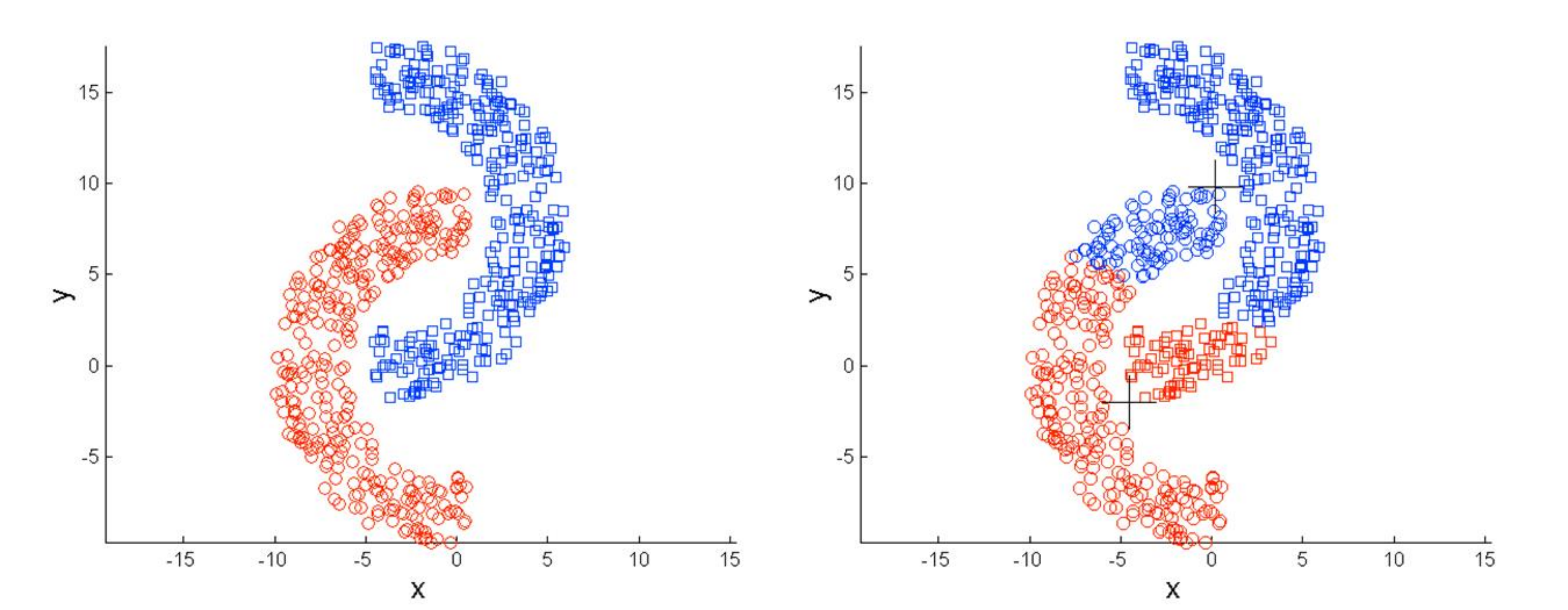
K-Means Clustering

- Limitations of K-Means Clustering
  - ✓ Cannot cope with different densities



K-Means Clustering

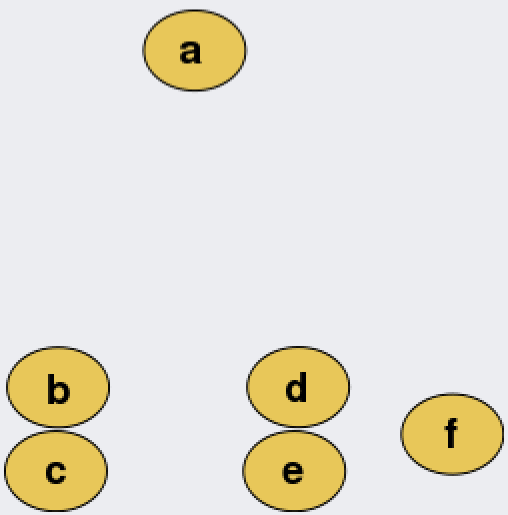
- Limitations of K-Means Clustering
  - ✓ Cannot cope with non-globular shapes

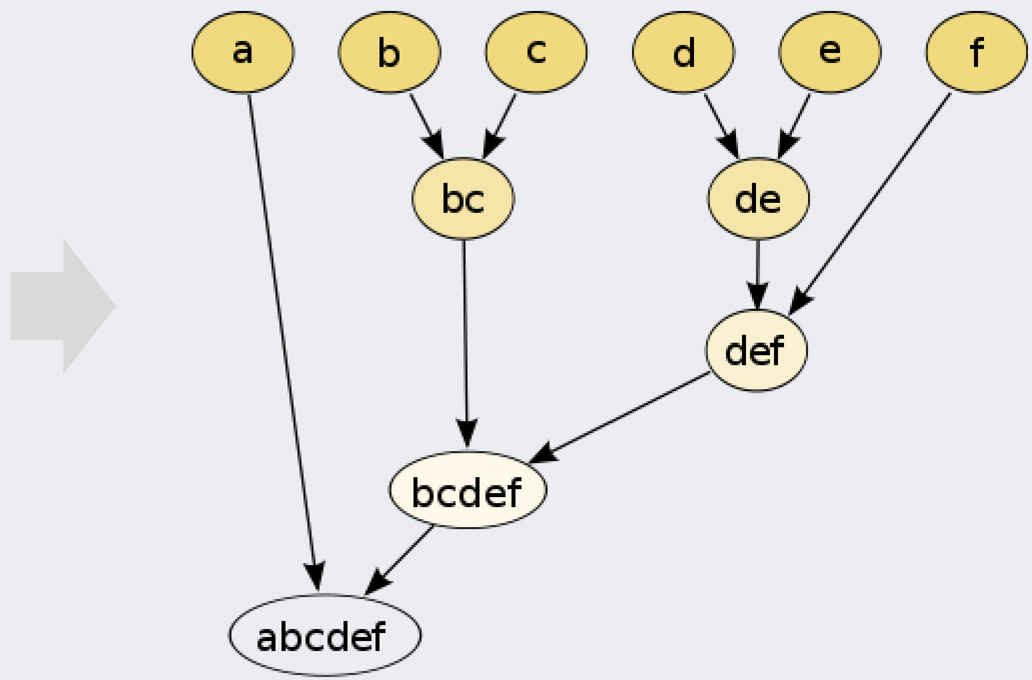


AGENDA

Hierarchical Clustering

- Hierarchical clustering
  - ✓ Produces a set of nested clusters organized as a hierarchical tree
  - ✓ Can be visualized as a dendrogram
    - A tree like diagram that records the sequences of merges or splits

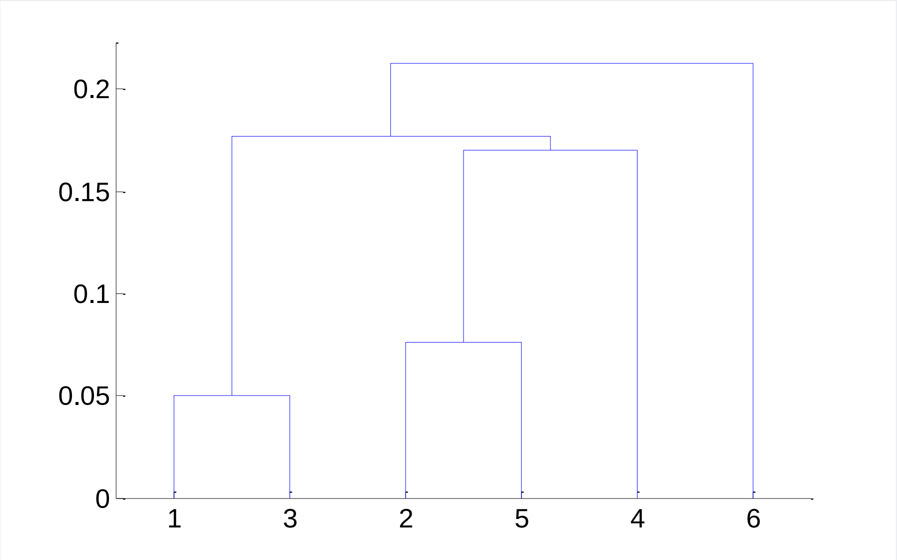


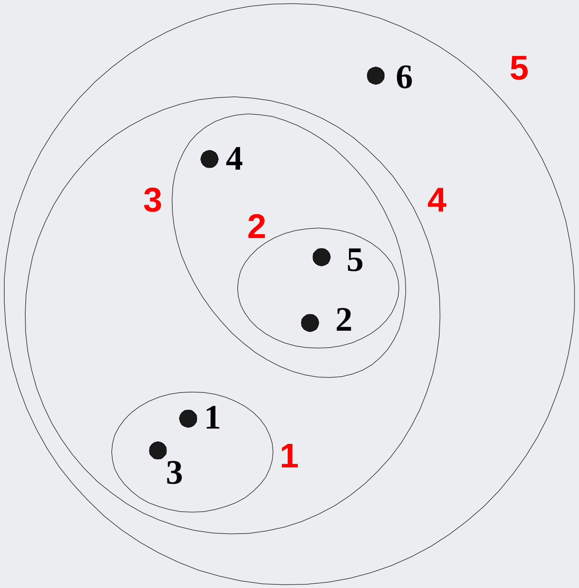


https://en.wikipedia.org/wiki/Hierarchical_clustering

Hierarchical Clustering

- Hierarchical clustering
  - ✓ Produces a set of nested clusters organized as a hierarchical tree
  - ✓ Can be visualized as a dendrogram
    - A tree like diagram that records the sequences of merges or splits



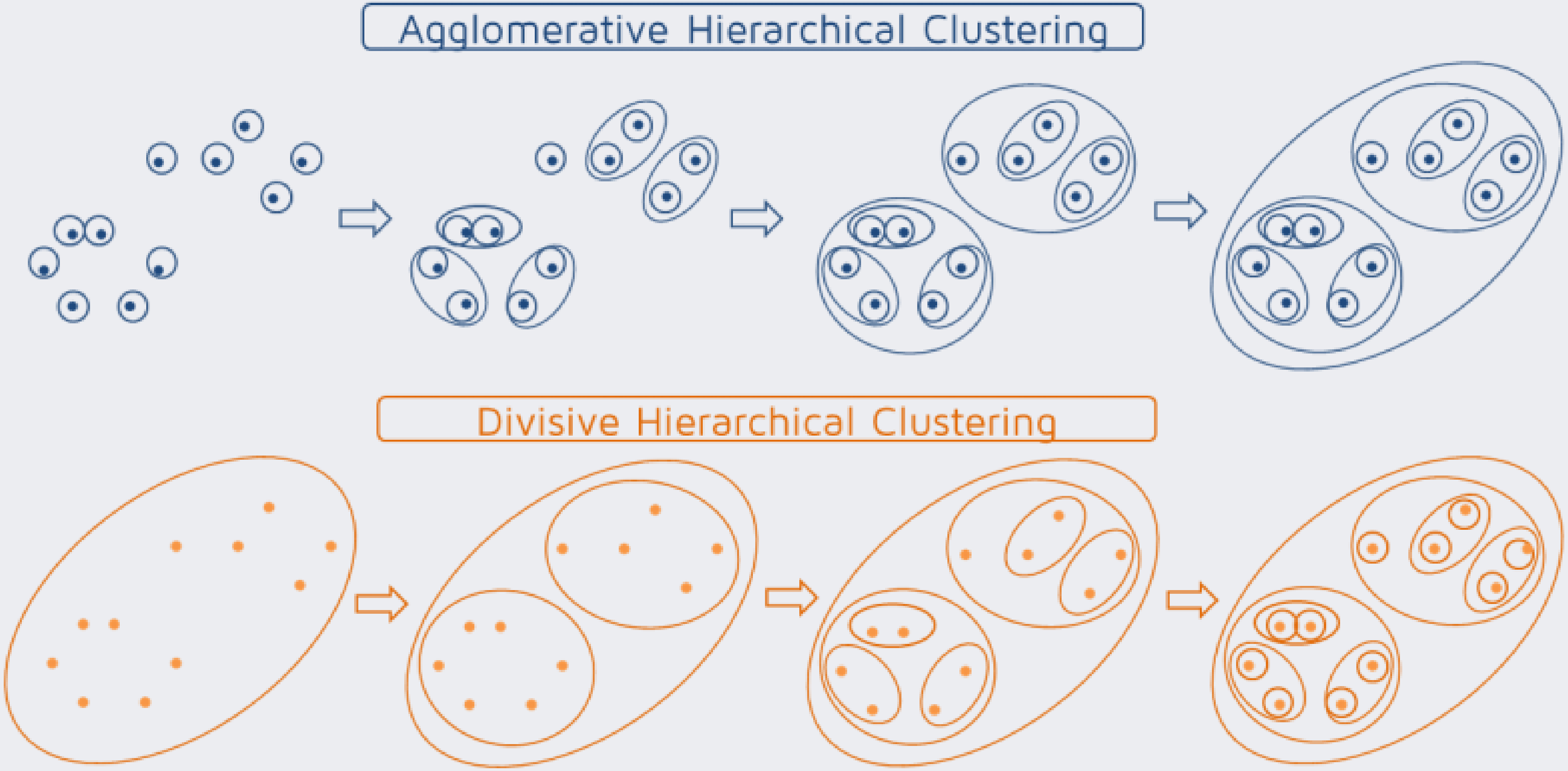


Hierarchical Clustering

- Strengths of Hierarchical clustering
  - ✓ Do not have to assume any particular number of clusters
    - Any desired number of clusters can be obtained by **'cutting'** the dendrogram at the proper level
  - ✓ May correspond to meaningful taxonomies
- Two main types of hierarchical clustering
  - ✓ Agglomerative clustering
    - Start with the points as individual clusters
    - At each step, merge the closest pair of clusters until only one cluster left
  - ✓ Divisive clustering
    - Start with one, all-inclusive cluster
    - At each step, split a cluster until each cluster contains a point

Hierarchical Clustering

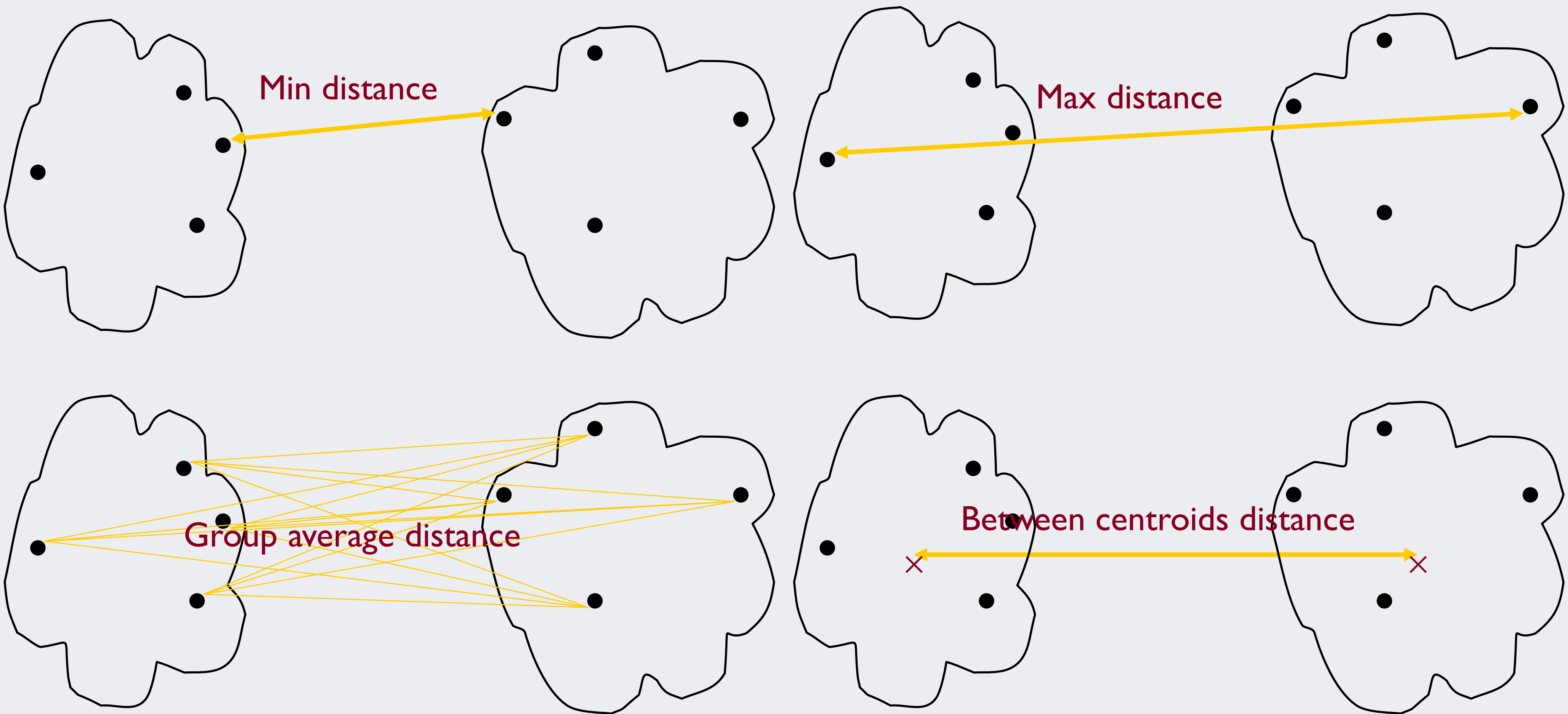
- Strengths of Hierarchical clustering
  - ✓ Agglomerative clustering vs. Divisive clustering



https://quantdare.com/hierarchical-clustering/

Hierarchical Clustering

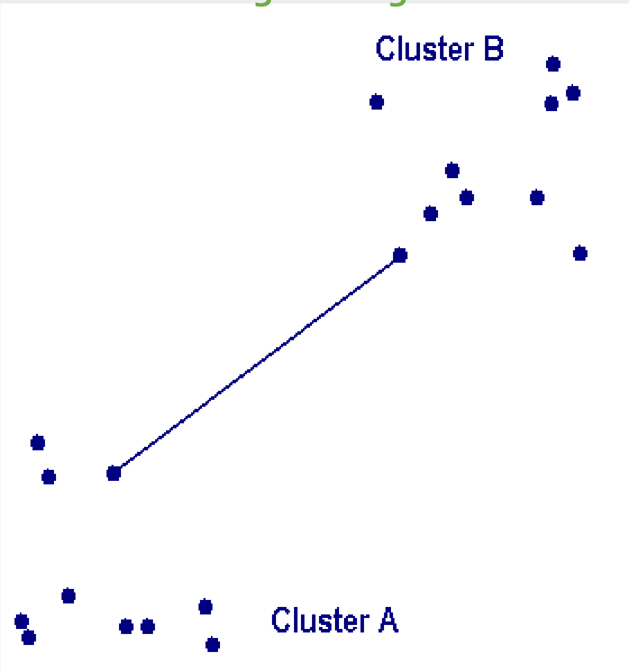
- Agglomerative clustering algorithm
  - ✓ Key operation: computation of the proximity of two clusters
    - Min, max, group average, between centroid, etc.



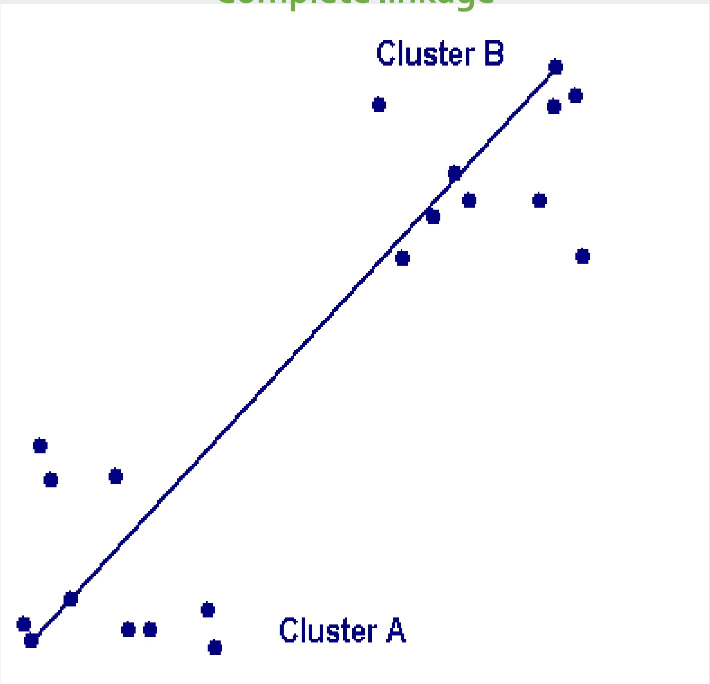
Hierarchical Clustering

- Agglomerative clustering algorithm
  - ✓ Single linkage: minimum distance between two data points in different clusters
  - ✓ Complete linkage: maximum distance between two data points in different clusters

Single linkage



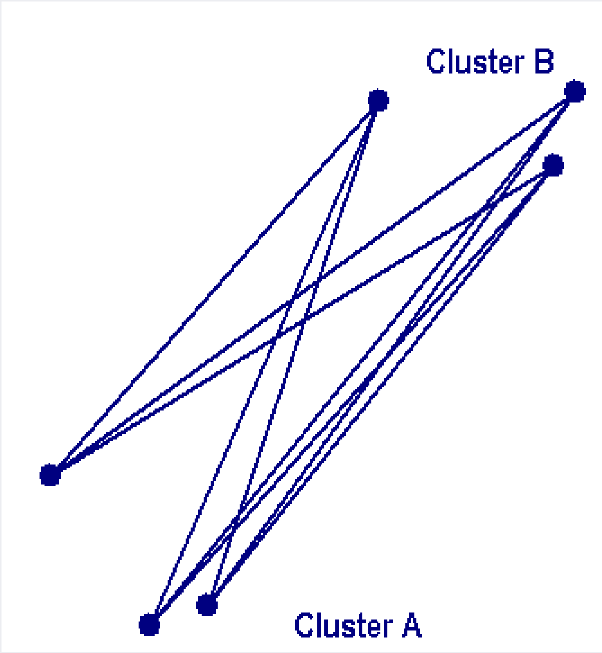
Complete linkage



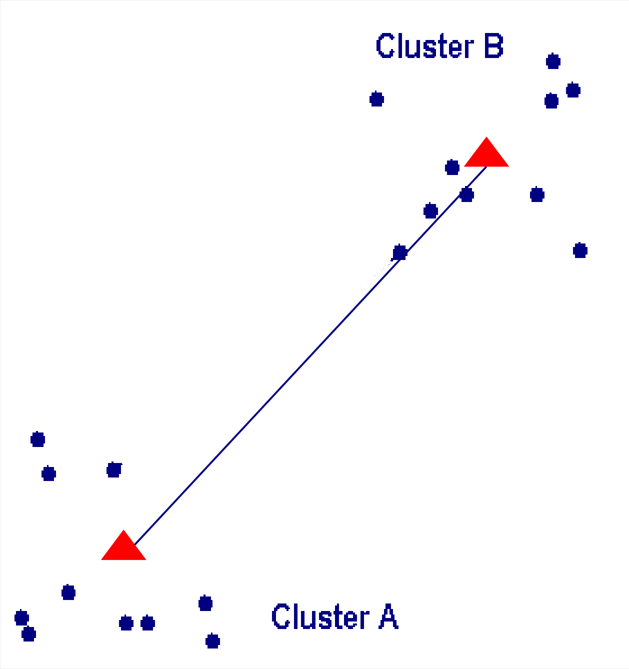
Hierarchical Clustering

- Agglomerative clustering algorithm
  - ✓ Average linkage: mean distance between two data points in different clusters
  - ✓ Centroid linkage: distance between centroids in different clusters

Average linkage

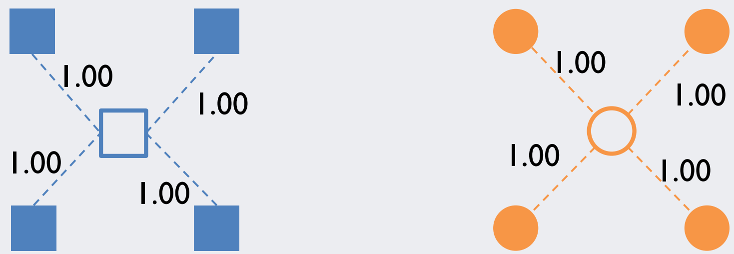


Centroid linkage

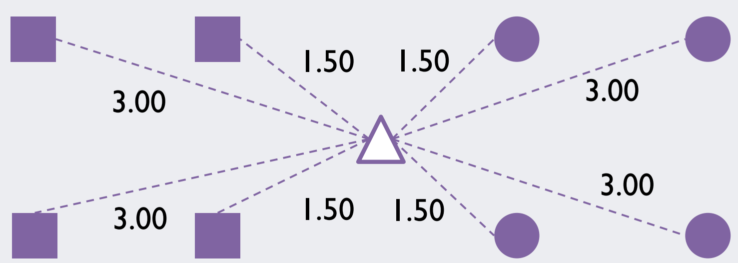


Hierarchical Clustering

- Agglomerative clustering algorithm
  - ✓ Ward method: Compare the sum of squared error (SSE) before and after the merge
    - SSE before merge: $1^2 + 1^2 + 1^2 + 1^2 + 1^2 + 1^2 + 1^2 + 1^2 = 8$



    - SSE after merge: $4 \times 1.5^2 + 4 \times 3^2 = 45$



    - Ward distance: 45-8 = 37

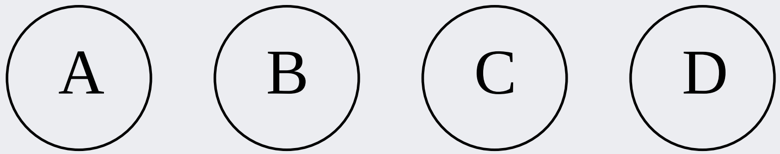
Hierarchical Clustering

- Agglomerative Clustering Procedure
  - ✓ Step 1: Assume that each data point is an individual cluster, compute the cluster distance
  - ✓ Step 2: Repeat the following procedure
    - Step 2-1: Merge the two closest clusters
    - Step 2-2: Update the cluster distance matrix
  - ✓ When all data points are merged as a single cluster, stop

Hierarchical Clustering

- Example

Initial Data Items



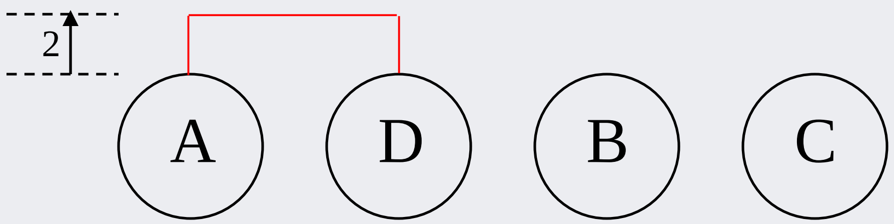
Distance Matrix

Dist	A	B	C	D
A		20	7	2
B			10	25
C				3
D				

Hierarchical Clustering

- Example

Current Clusters



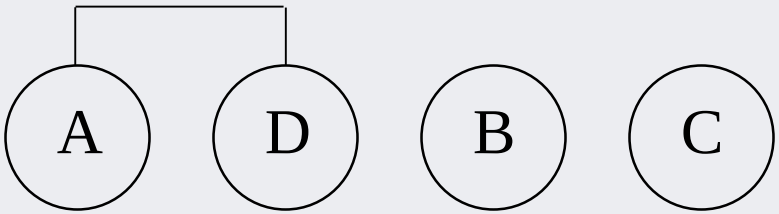
Distance Matrix

Dist	A	B	C	D
A		20	7	2
B			10	25
C				3
D				

Hierarchical Clustering

- Example

Current Clusters



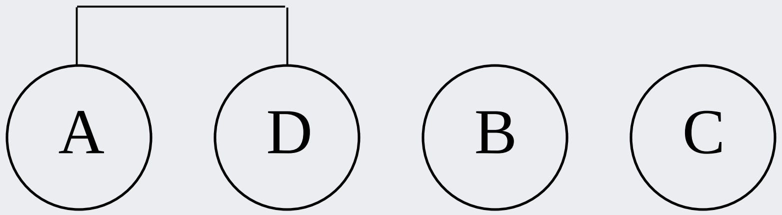
Distance Matrix

Dist	AD	B	C	
AD		20	3	
B			10	
C				

Hierarchical Clustering

- Example

Current Clusters



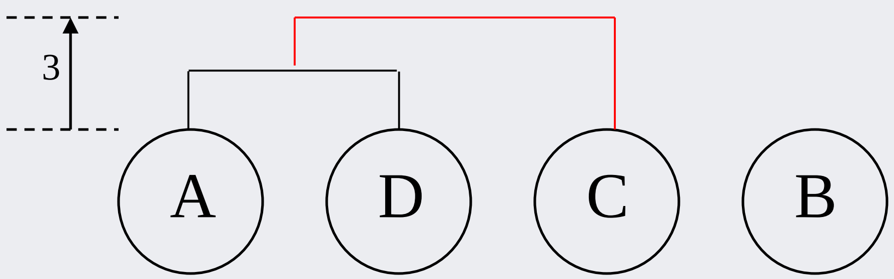
Distance Matrix

Dist	AD	B	C	
AD		20	3	
B			10	
C				

Hierarchical Clustering

- Example

Current Clusters



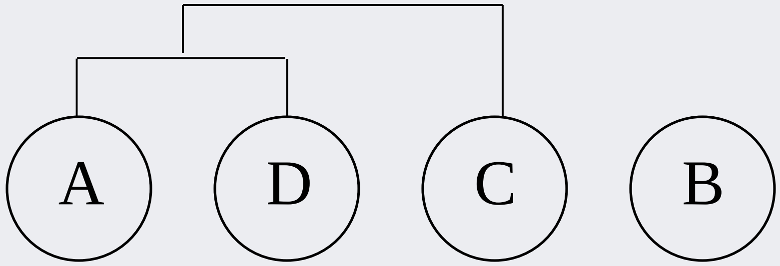
Distance Matrix

Dist	AD	B	C	
AD		20	3	
B			10	
C				

Hierarchical Clustering

- Example

Current Clusters



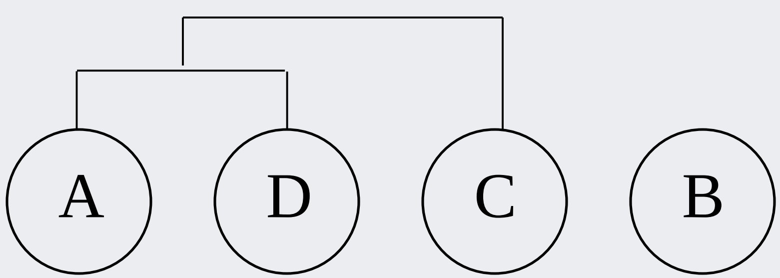
Distance Matrix

Dist	AD C	B		
AD C		10		
B				

Hierarchical Clustering

- Example

Current Clusters



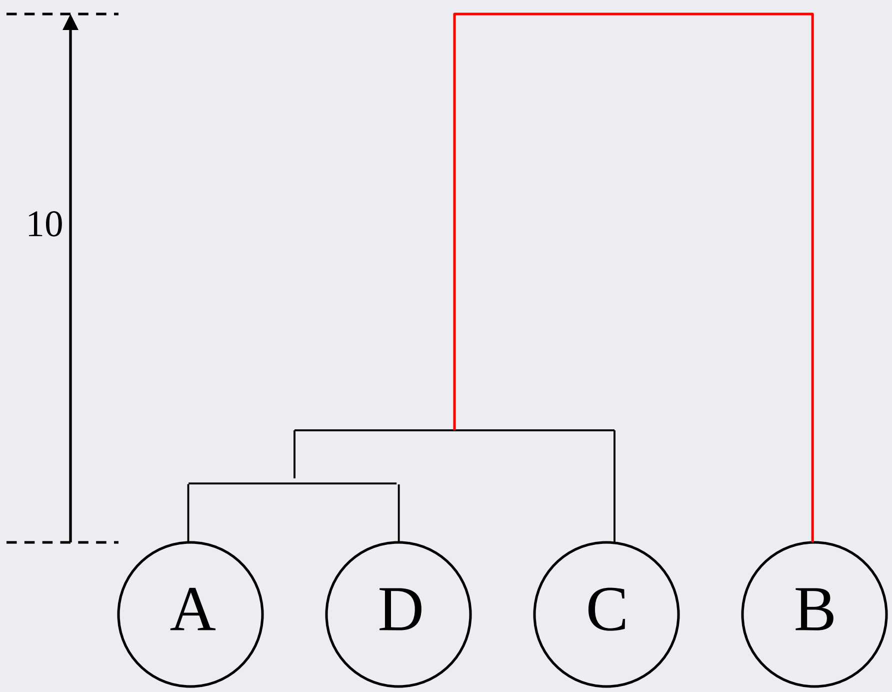
Distance Matrix

Dist	AD C	B		
AD C		10		
B				

Hierarchical Clustering

- Example

Current Clusters



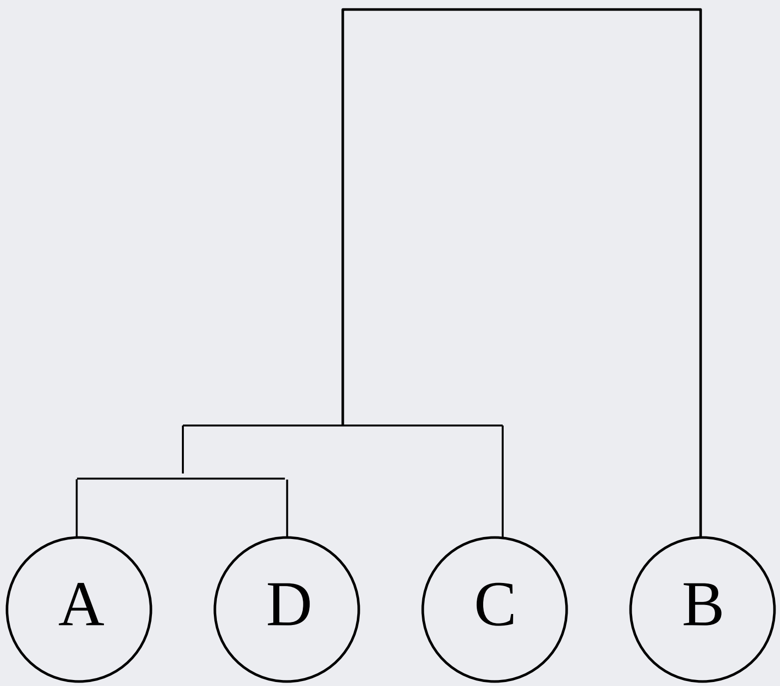
Distance Matrix

Dist	AD C	B		
AD C		10		
B				

Hierarchical Clustering

- Example

Final Result

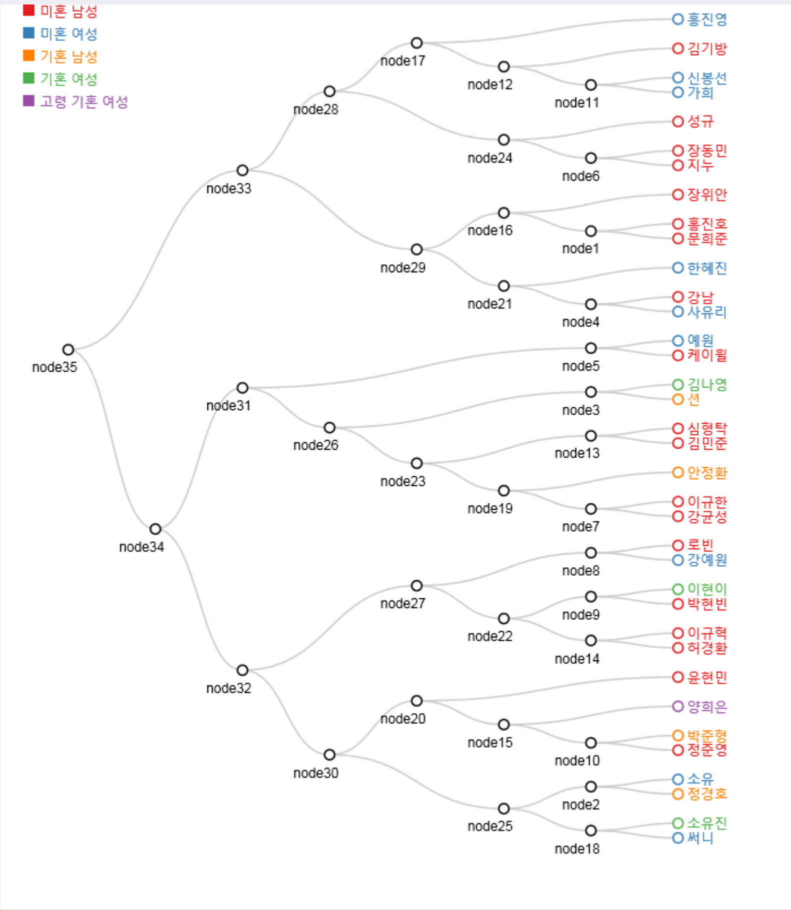


Distance Matrix

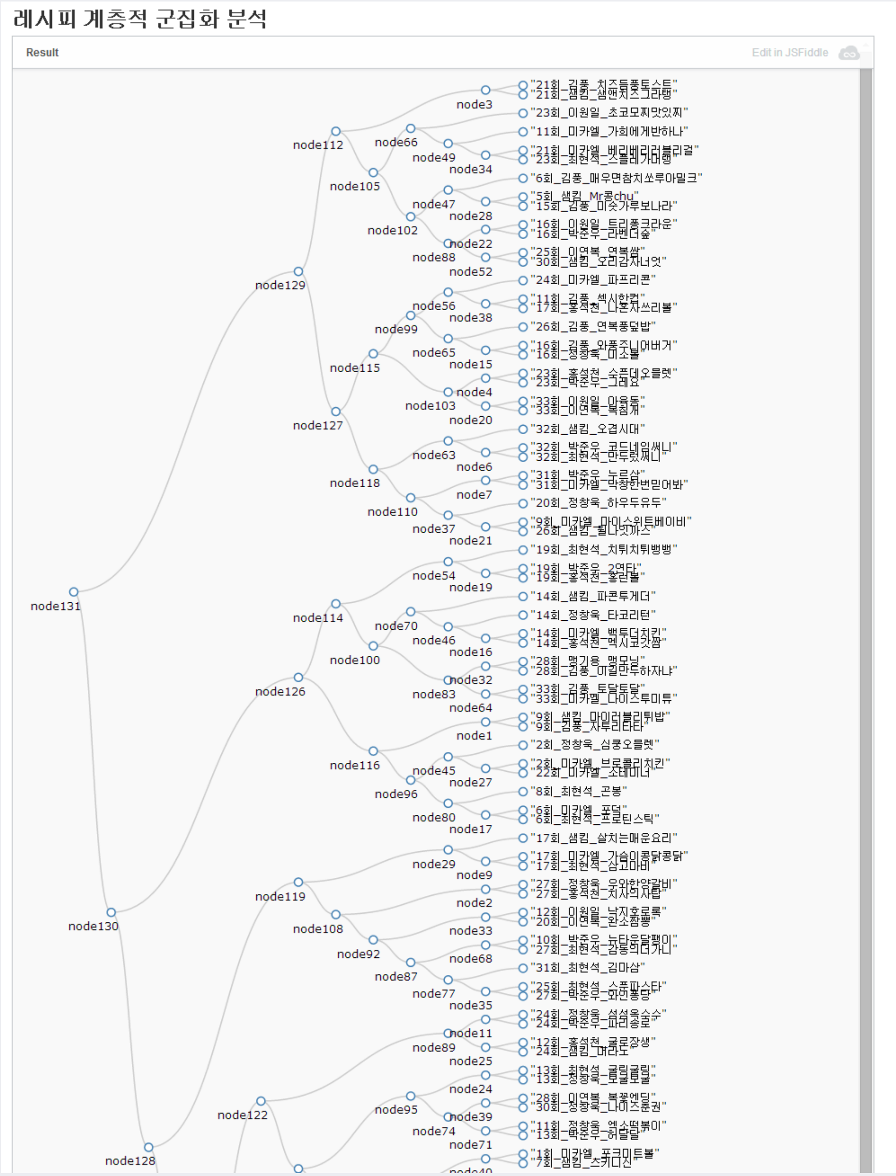
Dist	AD CB			
AD CB				

Hierarchical Clustering

- 냉장고를 부탁해!



냉장고 재료를 이용한 게스트 군집화



AGENDA

R Exercise: K-Means Clustering

- Dataset: Wine dataset from UCI machine learning repository

UCI

Machine Learning Repository

Center for Machine Learning and Intelligent Systems

Wine Data Set

Download: Data Folder, Data Set Description

Abstract: Using chemical analysis determine the origin of wines

Data Set Characteristics:	Multivariate	**Number of Instances:**	178	**Area:**	Physical
Attribute Characteristics:	Integer, Real	**Number of Attributes:**	13	**Date Donated**	1991-07-01
Associated Tasks:	Classification	**Missing Values?**	No	**Number of Web Hits:**	733901

Source:

Original Owners:

Forina, M. et al, PARVUS -
An Extendible Package for Data Exploration, Classification and Correlation.
Institute of Pharmaceutical and Food Analysis and Technologies, Via Brigata Salerno,
16147 Genoa, Italy.

Donor:

Stefan Aeberhard, email: stefan '@' coral.cs.jcu.edu.au

R Exercise: K-Means Clustering

- Dataset: Wine dataset from UCI machine learning repository
  - ✓ Dependent variable (column 1): grape species
  - ✓ Independent variables (column 2-14)
    - 1) Alcohol
    - 2) Malic acid
    - 3) Ash
    - 4) Alcalinity of ash
    - 5) Magnesium
    - 6) Total phenols
    - 7) Flavanoids
    - 8) Nonflavanoid phenols
    - 9) Proanthocyanins
    - 10) Color intensity
    - 11) Hue
    - 12) OD280/OD315 of diluted wines
    - 13) Proline

R Exercise: K-Means Clustering

- Install and load necessary packages

```r
# Package for cluster validity
install.packages("clValid")
install.packages("plotrix")
library(clValid)
library(plotrix)
```

  - ✓ "clValid" package
    - Provide functions to compute various cluster validity measures
  - ✓ "plotrix" package
    - Radar chart

- Many packages provide K-Means Clustering algorithm
  - ✓ stats, kml, kml3d, RSKC, skmeans, sparcl, etc.
  - ✓ In this exercise, we use the base function provided by R

R Exercise: K-Means Clustering

- Data load and preprocessing

```
# Part 1: K-Means Clustering ----------------------------------------------
# Load the Wine dataset
wine <- read.csv("wine.csv")

# Remove the class label
wine_class <- wine[,1]
wine_x <- wine[,-1]

# data scaling
wine_x_scaled <- scale(wine_x, center = TRUE, scale = TRUE)
```

✓ Data preprocessing

- Remove the first column because it is the dependent variable (clustering does not use the dependent variable)
- Data normalization to remove the impact of difference measurement units in dependent variables

R Exercise: K-Means Clustering

- Find the optimal number of clusters based on cluster validity measures

```r
# Evaluating the cluster validity measures
wine_clValid <- clValid(wine_x_scaled, 2:10, clMethods = "kmeans",
                        validation = c("internal", "stability"))
summary(wine_clValid)
```

✓ clValid( ): function that compute various cluster validity measures

- Arg 1: Dataset
- Arg 2: Candidate number of clusters (2 to 9 in this exercise)
- Arg 3: Clustering algorithm
- Arg 4: Category of cluster validity measures

R Exercise: K-Means Clustering

- Find the optimal number of clusters based on cluster validity measures
  - ✓ The optimal number of cluster is 3 based on the Dunn index and Silhouette index

```
> summary(wine_clValid)

Clustering Methods:
 kmeans

Cluster sizes:
 2 3 4 5 6 7 8 9 10

Validation Measures:
                            2       3       4       5       6       7       8        9       10

kmeans APN             0.1255  0.0470  0.1851  0.1392  0.1178  0.1634  0.2325   0.2543   0.2304
       AD              4.2577  3.6137  3.6186  3.4572  3.3486  3.3064  3.2889   3.2236   3.1174
       ADM             0.6454  0.2231  0.7547  0.5135  0.4513  0.5565  0.7912   0.8512   0.7698
       FOM             0.9139  0.7842  0.7728  0.7548  0.7435  0.7424  0.7264   0.7329   0.7157
       Connectivity   37.6512 28.0504 61.1659 76.2976 84.5433 88.4012 95.2159 109.1425 111.7302
       Dunn            0.1357  0.2323  0.1621  0.1900  0.2021  0.2111  0.2081   0.2081   0.2307
       Silhouette      0.2593  0.2849  0.2127  0.2656  0.2446  0.2323  0.2307   0.2210   0.2209

Optimal Scores:

             Score   Method Clusters
APN           0.0470 kmeans 3
AD            3.1174 kmeans 10
ADM           0.2231 kmeans 3
FOM           0.7157 kmeans 10
Connectivity 28.0504 kmeans 3
Dunn          0.2323 kmeans 3
Silhouette    0.2849 kmeans 3
```

R Exercise: K-Means Clustering

- Run KMC with the optimal number of clusters

```
# Perform K-Means Clustering with the best K determined by Silhouette
wine_kmc <- kmeans(wine_x_scaled,3)

str(wine_kmc)
wine_kmc$centers
wine_kmc$size
wine_kmc$cluster
```

✓ kmeans( ): KMC function

- Arg 1: Data set
- Arg 2: Number of clusters

✓ Output of kmeans( )

- $centers: coordinates of each centroids
- $size: the number of instances belonging to the cluster
- $cluster: cluster index

R Exercise: K-Means Clustering

- Compare the actual class and assigned clusters

```r
# Compare the cluster info. and class labels
real_class <- wine_class
kmc_cluster <- wine_kmc$cluster
table(real_class, kmc_cluster)
```

```
> table(real_class, kmc_cluster)
          kmc_cluster
real_class  1  2  3
         1  0  0 59
         2 65  3  3
         3  0 48  0
```

✓ Although we do not use the class information when doing KMC

- The optimal number of clusters is determined as 3 (= the number of classes)
- Each cluster is fairly homogeneous

R Exercise: K-Means Clustering

- Compare the actual class and assigned clusters

```
# Compare each cluster for KMC
cluster_kmc <- data.frame(wine_x_scaled, clusterID = as.factor(wine_kmc$cluster))
kmc_summary <- data.frame()

for (i in 1:(ncol(cluster_kmc)-1)){
        kmc_summary = rbind(kmc_summary, tapply(cluster_kmc[,i],
                            cluster_kmc$clusterID, mean))
}
colnames(kmc_summary) <- paste("cluster", c(1:3))
rownames(kmc_summary) <- colnames(wine_x)
kmc_summary
```

✓ Create a dataframe using the normalized data and cluster membership information

✓ kmc_summary

- Store the average value of each cluster
- Column name: cluster 1, cluster 2, cluster 3
- Row name: name of independent variables

R 실습: K-Means Clustering

- Compare the clusters

```
> kmc_summary
                              cluster 1   cluster 2  cluster 3
Alcohol.2                   -0.92346686  0.16444359  0.8328826
Malic.acid.                 -0.39293312  0.86909545 -0.3029551
Ash.                        -0.49312571  0.18637259  0.3636801
Alcalinity.of.ash.           0.17012195  0.52289244 -0.6084749
Magnesium.                  -0.49032869 -0.07526047  0.5759621
Total.phenols.              -0.07576891 -0.97657548  0.8827472
Flavanoids.                  0.02075402 -1.21182921  0.9750690
Nonflavanoid.phenols        -0.03343924  0.72402116 -0.5605085
Proanthocyanins.             0.05810161 -0.77751312  0.5786543
Color.intensity.            -0.89937699  0.93889024  0.1705823
Hue                          0.46050459 -1.16151216  0.4726504
OD280.OD315.of.diluted.wines. 0.27000254 -1.28877614  0.7770551
Proline.                    -0.75172566 -0.40594284  1.1220202
```

✓ Alcohol: Cluster 3 > Cluster 2 > Cluster 1

✓ Malic acid: Cluster 2 is the highest, little difference between Cluster 1 and 3

✓ Flavornoids: Cluster 3 > Cluster 1 > Cluster 2, differences are significant

R 실습: K-Means Clustering

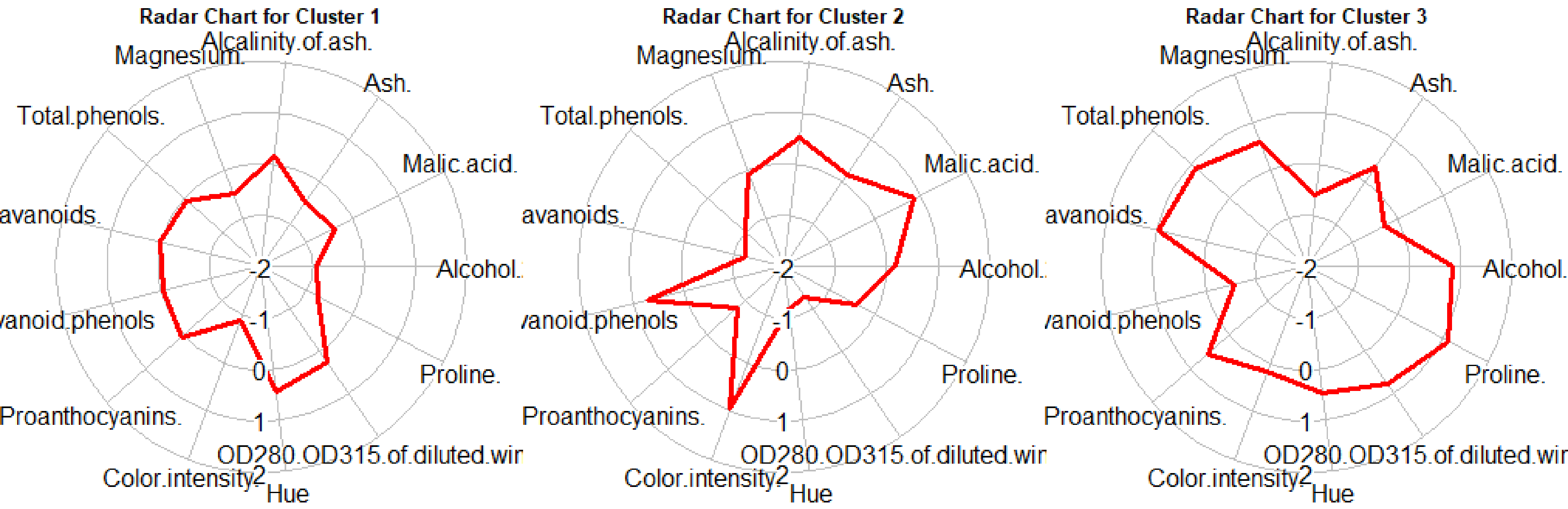
- Radar chart based on the average value of all variables

```
# Radar chart
par(mfrow = c(1,3))
for (i in 1:3){
         plot_title <- paste("Radar Chart for Cluster", i, sep=" ")
         radial.plot(kmc_summary[,i], labels = rownames(kmc_summary),
                           radial.lim=c(-2,2), rp.type = "p", main = plot_title,
                           line.col = "red", lwd = 3, show.grid.labels=1)
}
dev.off()
```

  - ✓ par(mfrow = c(1,3)): initialize 1 by 3 plots
  - ✓ radial.plot( ): draw a radar chart
  - ✓ dev.off( ): reset the parameters

R 실습: K-Means Clustering

- Radar chart based on the average value of all variables



R 실습: K-Means Clustering

- Hypothesis test for Cluster 1 and 2
  - ✓ H0: Average of Cluster 1 = Average of Cluster 2
  - ✓ H1:
    - First column: Average of Cluster 1 ≠ Average of Cluster 2
    - Second column: Average of Cluster 1 > Average of Cluster 2
    - Third column: Average of Cluster 1 < Average of Cluster 2

R 실습: K-Means Clustering

- Hypothesis test for Cluster 1 and 2

```
# Compare the first and the second cluster
kmc_cluster1 <- wine_x[wine_kmc$cluster == 1,]
kmc_cluster2 <- wine_x[wine_kmc$cluster == 2,]

# t_test_result
kmc_t_result <- data.frame()
for (i in 1:13){
         kmc_t_result[i,1] <- t.test(kmc_cluster1[,i], kmc_cluster2[,i],
                                     alternative = "two.sided")$p.value
         kmc_t_result[i,2] <- t.test(kmc_cluster1[,i], kmc_cluster2[,i],
                                     alternative = "greater")$p.value
         kmc_t_result[i,3] <- t.test(kmc_cluster1[,i], kmc_cluster2[,i],
                                     alternative = "less")$p.value
}
kmc_t_result
```

✓ t-test( ): function that perform the Student's t-test

- Arg 1 & 2: Variable values for cluster 1 & 2
- Arg 3: hypothesis ("two-sided,", "greater," "less")
- $p.value: significant value

R 실습: K-Means Clustering

- Hypothesis test for Cluster 1 and 2

```
> kmc_t_result
             V1           V2           V3
1  1.018635e-14 1.000000e+00 5.093173e-15
2  1.475032e-10 1.000000e+00 7.375160e-11
3  1.075500e-04 9.999462e-01 5.377501e-05
4  2.114410e-02 9.894280e-01 1.057205e-02
5  1.118367e-02 9.944082e-01 5.591835e-03
6  3.791205e-10 1.895603e-10 1.000000e+00
7  2.146292e-25 1.073146e-25 1.000000e+00
8  7.509477e-05 9.999625e-01 3.754739e-05
9  7.432607e-07 3.716303e-07 9.999996e-01
10 4.041358e-18 1.000000e+00 2.020679e-18
11 3.690316e-22 1.845158e-22 1.000000e+00
12 1.850143e-30 9.250715e-31 1.000000e+00
13 2.031718e-05 9.999898e-01 1.015859e-05
```

The average value of Cluster 1 is greater than the average value of Cluster 2 at the significance level of 0.01

The average value of Cluster 1 is smaller than the average value of Cluster 2 at the significance level of 0.01

R 실습: Hierarchical Clustering

- Data loading and preprocessing

```
# Part 2: Hierarchical Clustering -----------------------------------------
ploan <- read.csv("Personal Loan.csv")
ploan_x <- ploan[,-c(1,5,10)]
ploan_x_scaled <- scale(ploan_x, center = TRUE, scale = TRUE)
```

✓ Personal loan dataset

- Remove the first (ID-related variable) and fifth (zip code) variable
- The 10th column is not used because it is a dependent variable
- We must normalize the data because pairwise distance computation is involved

R 실습: Hierarchical Clustering

- Distance matrix

```
# Compute the similarity using the spearman coefficient
cor_Mat <- cor(t(ploan_x_scaled), method = "spearman")
dist_ploan <- as.dist(1-cor_Mat)
```

✓ Use correlation to compute the similarity between two instances

- t( ): transpose (change the rows and columns)
- cor( ): compute correlation

✓ as.dist( ): store the distance information more efficiently than the original distance matrix

- It stores upper or lower trangle values

R 실습: Hierarchical Clustering

- Perform HC

```
# Perform hierarchical clustering
hr <- hclust(dist_ploan, method = "complete", members=NULL)
```

✓ hclust( ): function for hierarchical clustering

- Arg 1: distance matrix
- Arg 2: distance computation methods between two clusters
- We do not need to set the number of clusters

R 실습: Hierarchical Clustering

- Dendrogram

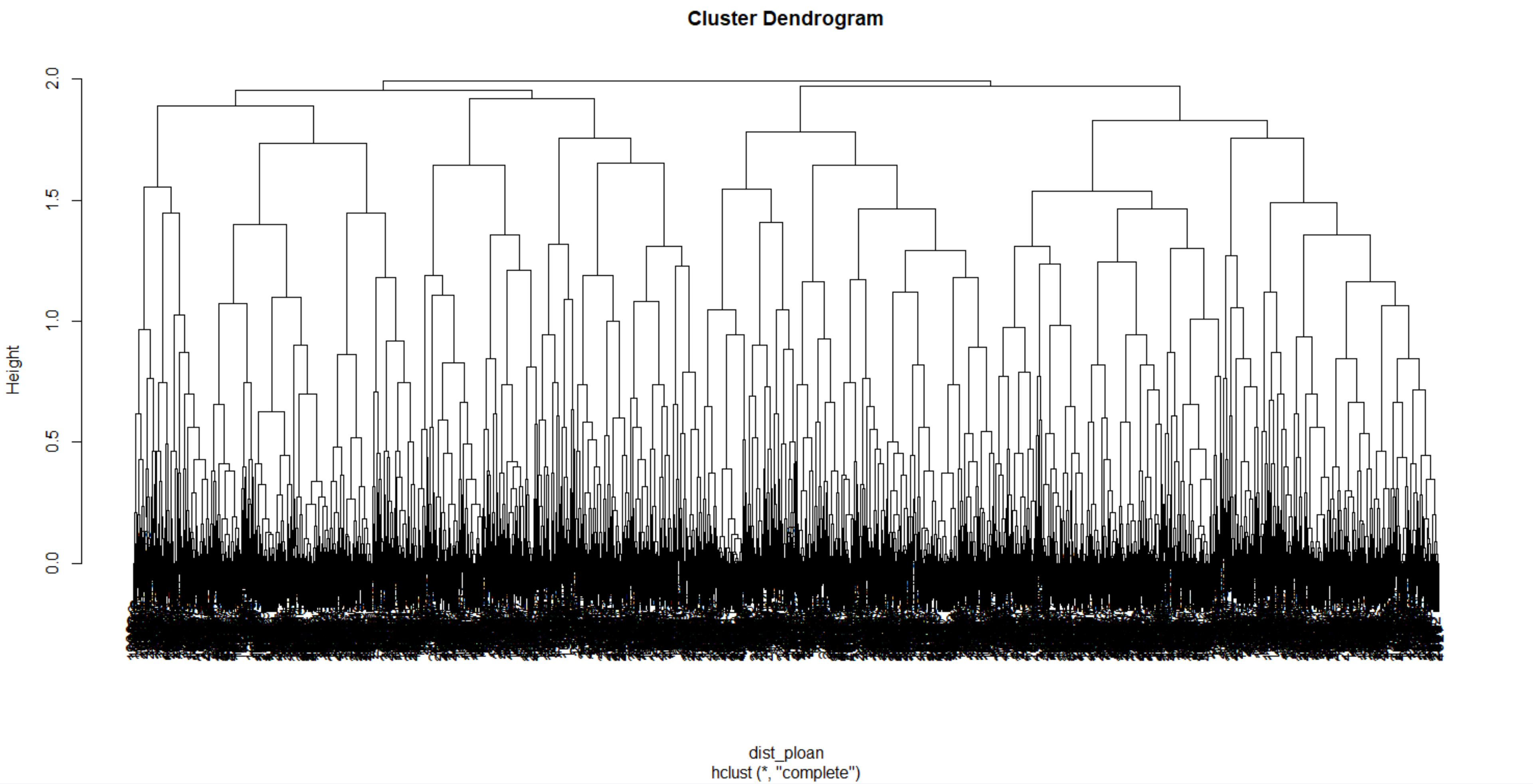
```r
# plot the results
plot(hr)
plot(hr, hang = -1)
plot(as.dendrogram(hr), edgePar=list(col=3, lwd=4), horiz=T)
```

✓ Options for drawing dendrogram

- Plot 1: basic style
- Plot 2: heights for the terminal nodes are different
- Plot 3: Rotated dendrogram

R 실습: Hierarchical Clustering

- Dendrogram



R 실습: Hierarchical Clustering

- Make clusters

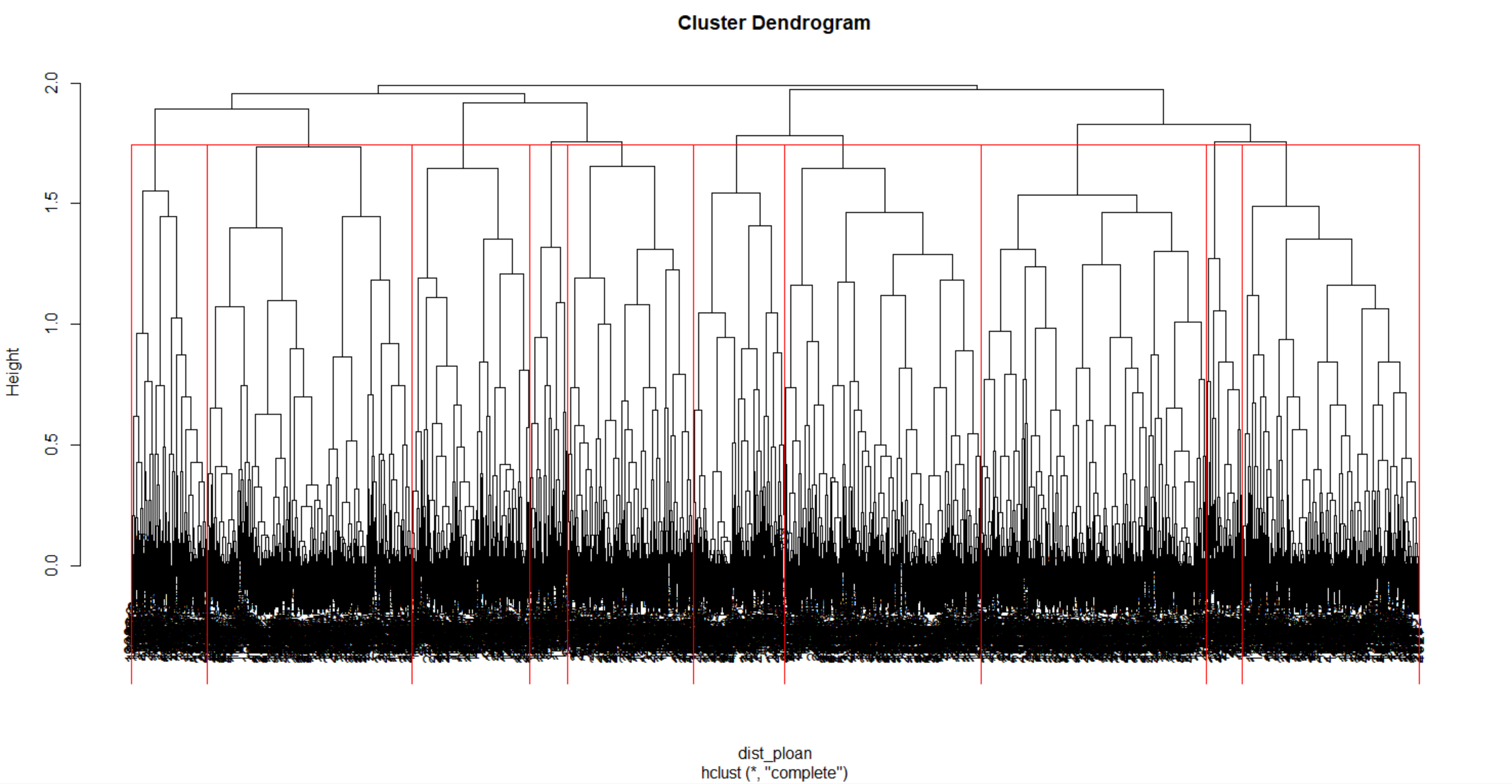
```
# Find the clusters
mycl <- cutree(hr, k=10)
mycl

plot(hr)
rect.hclust(hr, k=10, border="red")
```

- ✓ cutree( ): function that create clusters
  - Arg 1: dendrogram
  - Arg 2: the number of clusters
- ✓ rect.hclust( ): draw the rectangles for each cluster

R 실습: Hierarchical Clustering

- Make clusters



R 실습: Hierarchical Clustering

- Compare the clusters

```
# Compare each cluster for HC
ploan_hc <- data.frame(ploan_x_scaled, ploanYN = ploan[,10],
                            clusterID = as.factor(mycl))
hc_summary <- data.frame()
for (i in 1:(ncol(ploan_hc)-1)){
         hc_summary = rbind(hc_summary, tapply(ploan_hc[,i],
                            ploan_hc$clusterID, mean))
}
colnames(hc_summary) <- paste("cluster", c(1:10))
rownames(hc_summary) <- c(colnames(ploan_x), "LoanRatio")
hc_summary
```

✓ Summarize the clusters with the variable average values

✓ Compute the loan ratio for each cluster

  - Target marketing campaign should be offered to the cluster with the highest loan ratio
  - We can compare the two clusters (highest loan ratio vs. lowest loan ratio)

R 실습: Hierarchical Clustering

- Compare the clusters

```
> hc_summary
```

	cluster 1	cluster 2	cluster 3	cluster 4	cluster 5	cluster 6	cluster 7	cluster 8	cluster 9	cluster 10
Age	-0.978375630	0.62768747	-1.00065813	0.45556464	0.623411101	0.3606053	0.83631077	-0.94640912	0.23039152	-0.27857180
Experience	-0.984955242	0.63865510	-1.01149000	0.45826338	0.594577161	0.3583185	0.84354883	-0.93187619	0.22877326	-0.27319032
Income	-0.041720588	0.11353167	-0.34249916	0.67417711	0.055576655	0.2654431	-0.76794149	0.68688171	-0.30617789	0.99685044
Family	0.249749572	-0.06963678	0.64165613	-0.81485549	0.758429117	-0.2598958	0.09881633	-0.61816862	-0.84443724	-0.42263030
CCAvg	0.056049820	0.07592153	-0.27792518	0.55790241	0.002870789	0.3816039	-0.55877036	0.47138980	-0.43458985	0.01431525
Education	0.219734047	-0.33718946	0.21399718	-0.13166151	0.168847347	0.0920721	0.02249967	-0.20870083	0.82666013	-0.77272322
Mortgage	-0.109905334	-0.04520828	-0.09913763	0.04240734	0.412406634	-0.3300013	-0.16572200	0.12002376	0.89665284	1.20219168
Securities.Account	0.002768788	0.02273880	0.04403728	0.13838377	-0.175403903	-0.3507727	0.13171186	0.13469351	-0.12216694	-0.21924611
CD.Account	-0.197847765	-0.21063348	-0.07349502	-0.04132722	-0.229615630	0.2672237	0.03286815	0.48516412	-0.08077930	0.98776848
Online	-1.200618309	-1.20906773	0.76690876	0.81151795	0.624917820	0.3859185	0.51306132	0.67151852	-1.10189473	0.42941455
CreditCard	-0.044161882	-0.17087863	-0.13667044	-0.59150572	-0.639594361	1.5628656	-0.25222704	-0.05804595	0.11553475	1.26115872
LoanRatio	0.098837209	0.10755149	0.07874016	0.17467249	0.102739726	0.1147541	0.01005025	0.21348315	0.07142857	0.20547945

✓ In terms of loan ratio

- Cluster 7 has the lowest loan ratio (1.01%) and cluster 8 has the highest loan ratio (21.35%)

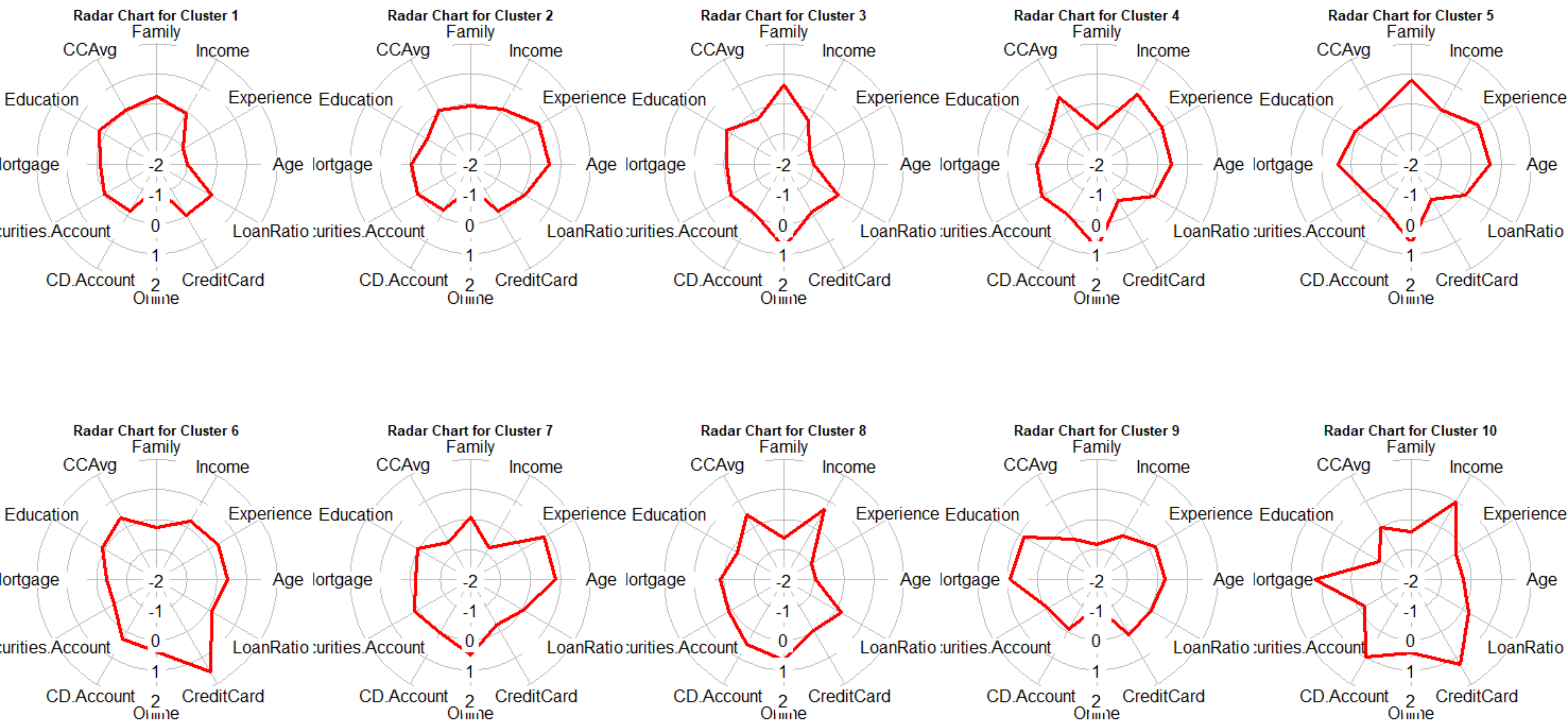
R 실습: Hierarchical Clustering

- Compare the clusters (Radar chart)

```
# Radar chart
par(mfrow = c(2,5))
for (i in 1:10){
         plot_title <- paste("Radar Chart for Cluster", i, sep=" ")
         radial.plot(hc_summary[,i], labels = rownames(hc_summary),
                                   radial.lim=c(-2,2), rp.type = "p", main = plot_title,
                                   line.col = "red", lwd = 3, show.grid.labels=1)
}
dev.off()
```

R 실습: Hierarchical Clustering

- Compare the clusters (Radar chart)



R 실습: Hierarchical Clustering

- Hypothesis test for Cluster 7 and 8
  - ✓ H0: Average of Cluster 7 = Average of Cluster 8
  - ✓ H1:
    - First column: Average of Cluster 7 ≠ Average of Cluster 8
    - Second column: Average of Cluster 7 > Average of Cluster 8
    - Third column: Average of Cluster 7 < Average of Cluster 8

R 실습: Hierarchical Clustering

- Hypothesis test for Cluster 7 and 8

```
# Compare the cluster 7 & 8
hc_cluster7 <- ploan_hc[ploan_hc$clusterID == 7, c(1:11)]
hc_cluster8 <- ploan_hc[ploan_hc$clusterID == 8, c(1:11)]

# t_test_result
hc_t_result <- data.frame()
for (i in 1:11){
         hc_t_result[i,1] <- t.test(hc_cluster7[,i], hc_cluster8[,i],
                                    alternative = "two.sided")$p.value
         hc_t_result[i,2] <- t.test(hc_cluster7[,i], hc_cluster8[,i],
                                    alternative = "greater")$p.value
         hc_t_result[i,3] <- t.test(hc_cluster7[,i], hc_cluster8[,i],
                                    alternative = "less")$p.value
}
hc_t_result
```

R 실습: Hierarchical Clustering

- Hypothesis test for Cluster 7 and 8
  - ✓ Variables with the higher average value for the cluster 7: Age, Experience, Family, Education
  - ✓ Variable with the higher average value for the cluster 8: Income, CCAvg, Mortgage, CD.Account, Online, Creditcard

```
> hc_t_result
             V1           V2           V3
1  1.216994e-95 6.084971e-96 1.000000e+00
2  5.671113e-96 2.835557e-96 1.000000e+00
3  2.573588e-41 1.000000e+00 1.286794e-41
4  5.199496e-21 2.599748e-21 1.000000e+00
5  2.054088e-26 1.000000e+00 1.027044e-26
6  1.147233e-02 5.736163e-03 9.942638e-01
7  5.157287e-03 9.974214e-01 2.578643e-03
8  9.770747e-01 5.114626e-01 4.885374e-01
9  6.261203e-04 9.996869e-01 3.130601e-04
10 3.591460e-03 9.982043e-01 1.795730e-03
11 2.184769e-02 9.890762e-01 1.092384e-02
```


ANY
questions?