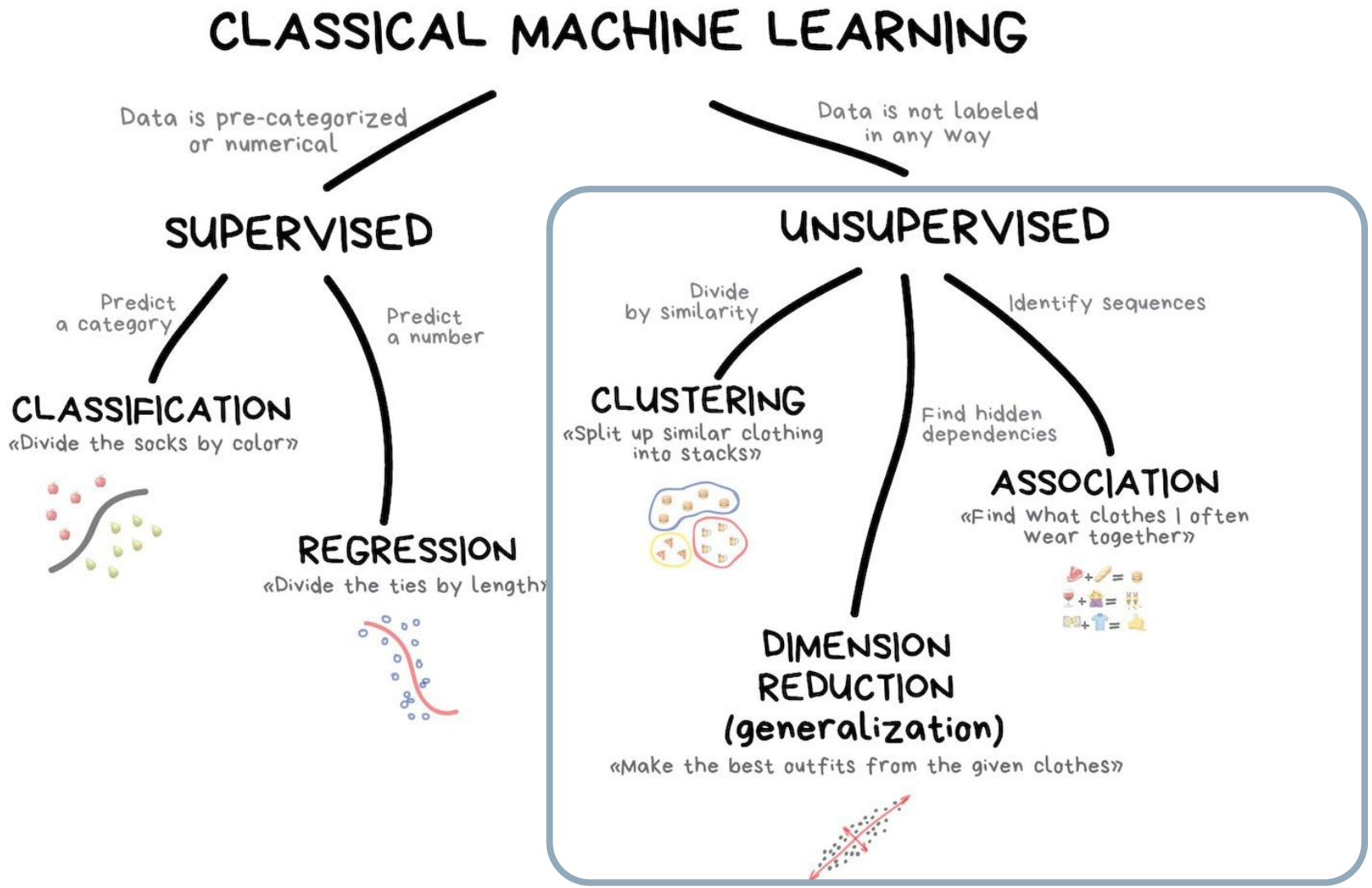
CLUSTERING

Week11

Type of Learning



Clustering

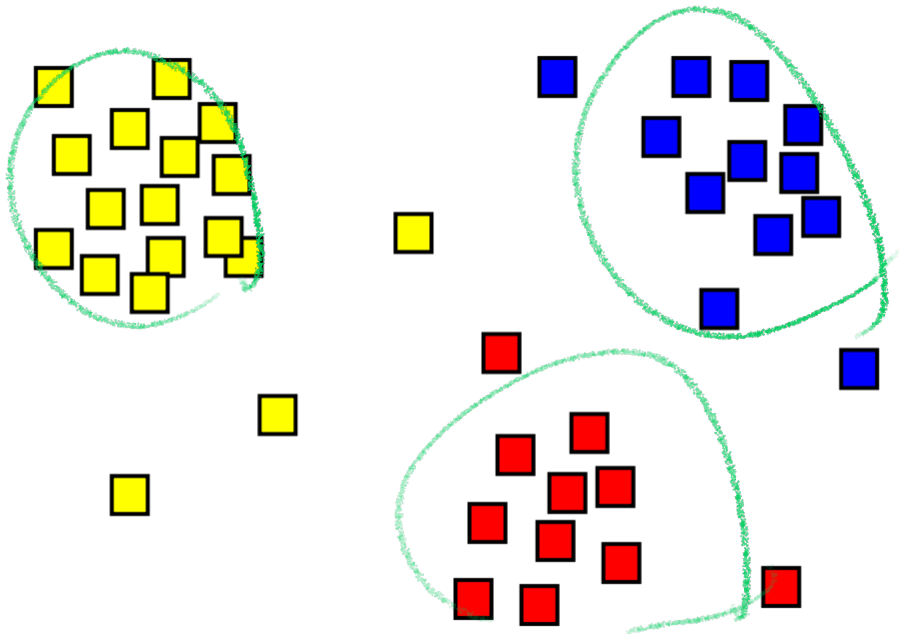
Unsupervised Learning: Clustering

- [Remind] Unsupervised learning is learning with unlabeled data
  - No certain output to be estimated

Clustering이란 데이터 포인트들을 다음 조건에 최대한 만족하도록 그룹화하는 것

- Clustering is to group a set of data points to satisfy following conditions as much as possible
  - Data points in the same group are more similar to each other than to data points in other groups

같은 그룹에 속한 데이터들은서로 더 유사하고 다른 그룹의 데이터들과는 덜 유사해야한다



Clustering

- Purpose
  - Discover patterns in data
  - Data compression and summarization
  - Anomaly detection
  - Preprocessing for other machine learning tasks

1. 데이터 안에 숨겨진 패턴이나 구조 발견
2. 데이터를 압축하거나 요약하는데 사용
3. 이상치 탐지
4. 다른 머신러닝 작업을 위한 전처리 과정

Clustering

- Types of clustering
  - Hard Clustering: Each data point belongs to only one cluster 각 데이터 포인트는 오직 하나의 클러스터에만 속함
  - Soft Clustering (Fuzzy Clustering): Each data point has a probability of belonging to multiple clusters 각 데이터 포인트는 여러 클러스터에 속할 확률을 가짐

- Types of clustering methods
  - Partition-based Clustering (e.g., K-Means): Partition-based clustering assigns data points into k clusters by minimizing intra-cluster variance. 데이터를 k개의 군집으로 나누되 군집 내부의 분산을 최소화하는 방식
  - Hierarchical Clustering: Builds a hierarchy of clusters using a tree-like structure (dendrogram) 트리구조(dendrogram)을 사용해 데이터 간 유사성에 따라 군집을 계층적으로 생성
  - Density-based Clustering (e.g., DBSCAN): Groups points that are closely packed together based on density 밀도가 높은 지점들을 군집으로 묶고 밀도가 낮은 부분은 노이즈로 간주
  - Model-based Clustering (e.g., Gaussian Mixture Models): Assumes that data is generated from a mixture of probability distributions
    데이터가 확률분포의 혼합에서 생성되었다고 가정

Clustering

- Data points in the same group are more similar to each other than to data points in other groups

1. How to know
some points are more similar than others?

⬇

Using distance measure

2. How to group?

⬇

Determine certain rule to group

k-means Clustering

k-means Clustering

- Objective function of clustering

$$\sum_i \min_j \|\mathbf{x}_i - \boldsymbol{\mu}_j\|^2$$

  - $j \in [1,2,\ldots,k]$
  - $\boldsymbol{\mu}_j$ is the centroid of j-th cluster

Combinatorial Optimization Problem

조합 최적화 문제:
클러스터링은 단순히 수학적으로 미분가능한 문제가 아니라 데이터를 k개의 그룹으로 나눈 조합을 선택해야하는 문제
이는 조합적인 성질을 가짐 정확한 해를 찾는 것은 NP-hard문제일 수 있다.
따라서 K-means는 정확한 최적해를 구하기보다는 반복적으로 갱신하면서 근사해를 찾음

k-means Clustering

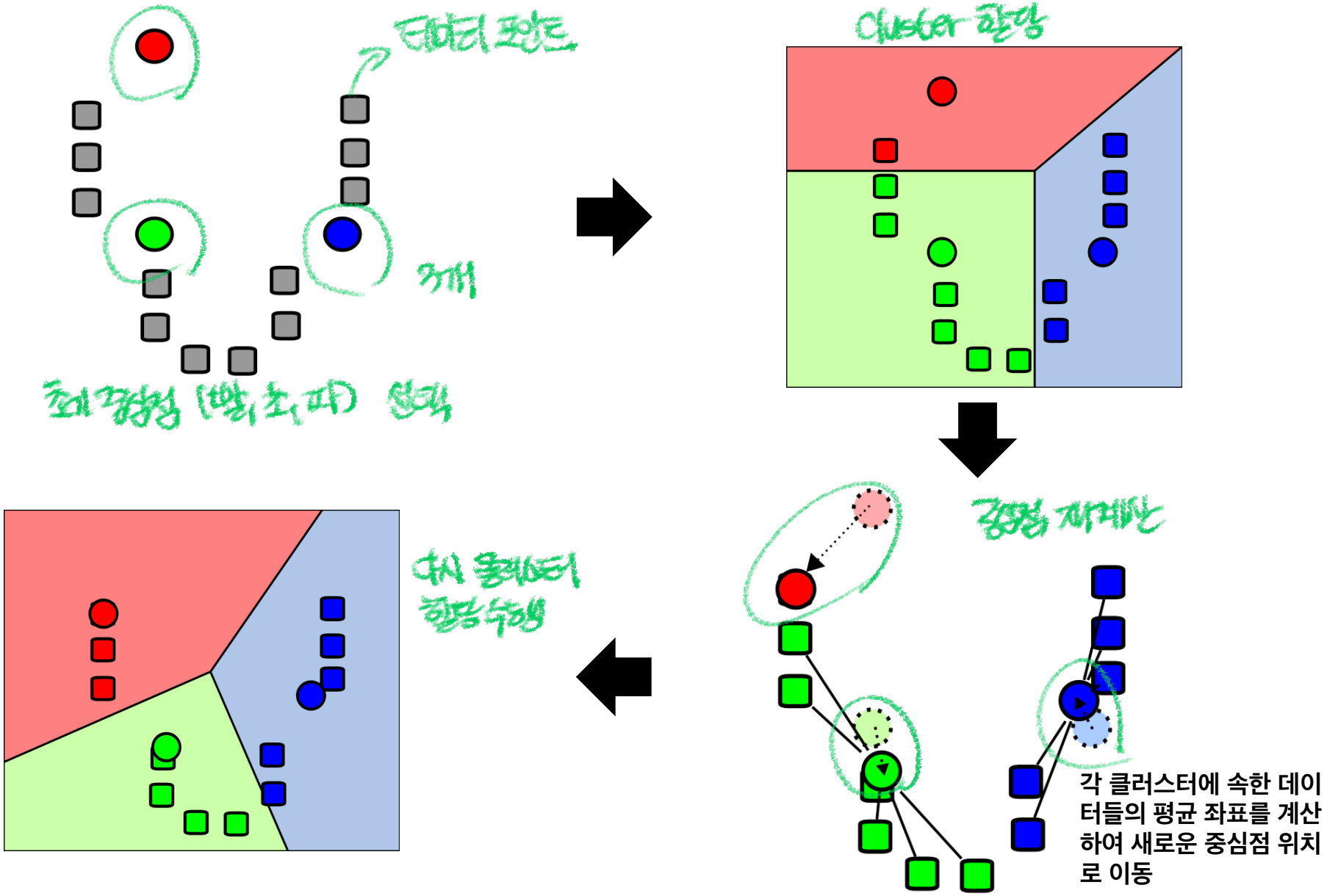
- Procedure of k-means clustering

1. • Set initial k centroids — 초기 중심점 k개 설정
2. • Assign each data point to i-th group whose centroid is the closest to data point among all centroids — 가장 중심점에 각 데이터 할당
3. • Update centroids — 각 클러스터에 속한 데이터들의 평균 좌표를 계산하여 중심점을 새로 갱신
4. • Check terminal condition and if condition is not fulfilled, go to step 2
   중심점이 더이상 변하지 않거나 반복 횟수가 기준값을 초과하면 종료, 그렇지 않으면 2단계로 할당

Terminal conditions

[No change in centroids or the number of iteration is over the pre-specified threshold]

k-means Clustering



How to Update Centroids

- Arithmetic mean

$$m_i^{(t+1)} = \frac{1}{\left|S_i^{(t)}\right|} \sum_{x_j \in S_i^{(t)}} x_j$$

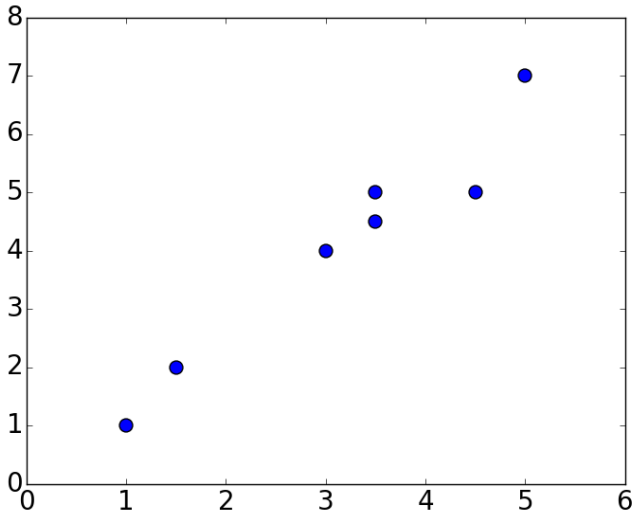
  - t is iteration
  - m_i is i-th group centroid
  - S_i is a set of i-th group and $|S_i|$ is size of S_i

- Example
  - If (3,1), (2,2), (4,6) belong to group, updated centroid is

$$\left(\frac{3+2+4}{3}, \frac{1+2+6}{3}\right) = (3, 2)$$

Question

- Clustering for 2D data set



	1	2	3	4	5	6	7
x	1.0	1.5	3.0	5.0	3.5	4.5	3.5
y	1.0	2.0	4.0	7.0	5.0	5.0	4.5

1) When $k = 2$ and initial centroids are (1.0, 1.0) and (5.0, 7.0), determine group of each data point

2) What are new centroids of two groups?

第0：(1.0, 1.0) (1.5, 2.0) (3.0, 4.0)

$$\left(\frac{1+1.5+3.0}{3}, \frac{1.0+2.0+4.0}{3}\right) = (1.83, 2.33)$$

第1：(5.0, 7.0) (3.5, 5.0) (4.5, 5.0) (3.5, 4.5)

$$\left(\frac{3.5+4.5+3.5}{4}, \frac{7+5+5+4}{4}\right) = (4.13, 5.38)$$

k-means Clustering: Pros and Cons

- Pros
  - Simple and efficient
  - Works well with large datasets

- Cons
  - Requires specifying k
  - Sensitive to initial centroid selection

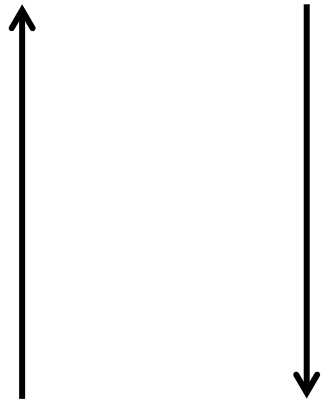
Hierarchical Clustering

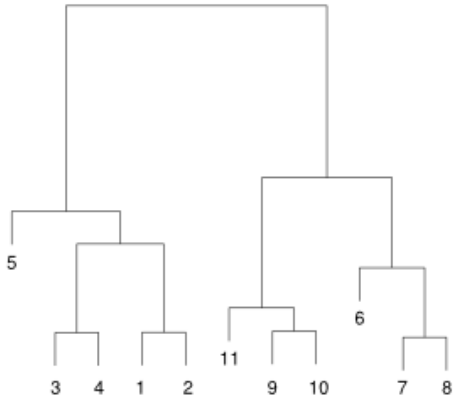
Hierarchical Clustering

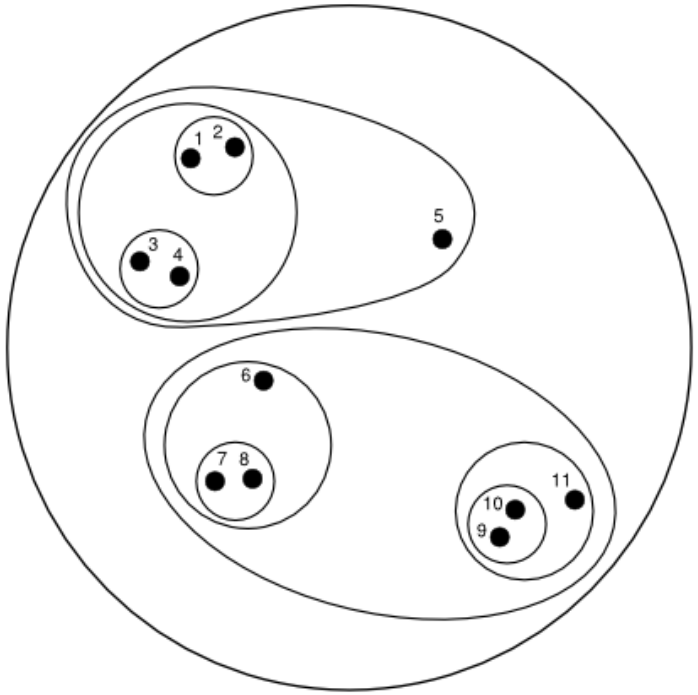
- Hierarchical clustering builds a hierarchy of clusters
  - Agglomerative: Bottom up approach, each data point starts in its own cluster and pairs of clusters are merged as one moves up the hierarchy
  - Divisive: Top down approach, all data points start in one cluster and splits are performed recursively as one moves down the hierarchy

모든 데이터가 하나의 클러스터로 시작 -> 조건에 따라 클러슽터를 반복적으로 분할 -> 원하는 수 만큼 클러스터가 나올 때까지 재귀적으로 수행

Divisive







Agglomerative

각 데이터 포인트가 처음에는 자시난의 클러스터로 시작 -> 가장 가까운 클러스트를 반복적으로 병합 -> 마지막에는 모든 포인트가 하나의 클러스터가 됨

Linkage Criteria for Agglomerative Clustering

- Way to calculate similarity between two clusters A, B

Type	Formula
Complete-linkage	$\max\{d(a,b): a \in A, b \in B\}$
Single-linkage	$\min\{d(a,b): a \in A, b \in B\}$
Mean linkage	$\frac{1}{\lvert A\rvert\lvert B\rvert}\sum_{a\in A}\sum_{b\in B} d(a,b)$
Centroid linkage	$d(c_A, c_B)$
Ward linkage	$\mathrm{Var}(A \cup B) - \mathrm{Var}(A) - \mathrm{Var}(B)$

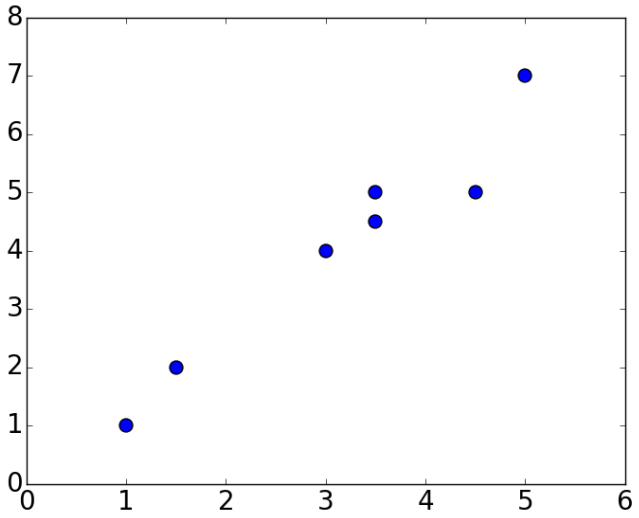
  - a belongs to A, b belongs to B
  - $\mathrm{Var}(X)$ is within-cluster variance (variance of cluster X)

$$Var(X) = \frac{1}{n_A}\sum_{i\in A}\|\mathbf{x}_i - \mu_A\|^2$$

  - $d(a,b)$ is distance between two data points a and b

Question

- Clustering for 2D data set



	1	2	3	4	5	6	7
x	1.0	1.5	3.0	5.0	3.5	4.5	3.5
y	1.0	2.0	4.0	7.0	5.0	5.0	4.5
C	1	1	2	2	2	2	2

1) Using complete-linkage, calculate linkage criterion of cluster 1 and 2

2) Using centroid-linkage, calculate linkage criterion of cluster 1 and 2

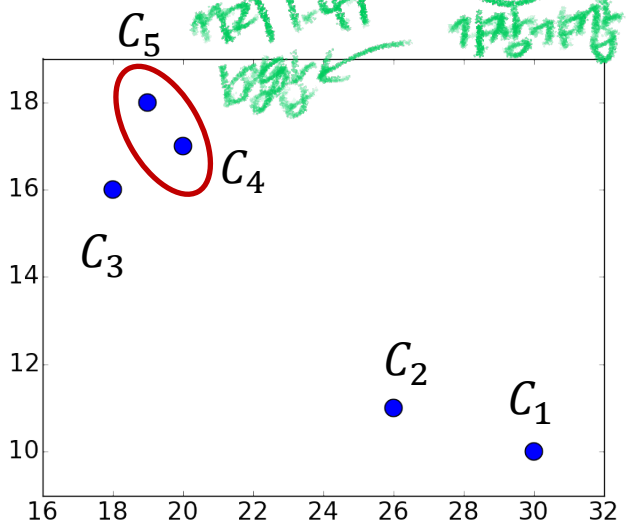
Example: Single Linkage Clustering

- Find clusters through single linkage hierarchy clustering
  - Start each data as own cluster
  - Distance measure between two points: Euclidean distance

Distance $d(A, B) = \min\{d(a, b): a \in A, b \in B\}$

	1	2	3	4	5
1	0				
2	4.12	0			
3	15.23	11.18	0		
4	12.21	8.48	4.12	0	
5	13.60	9.90	3.61	**1.41**	0

	1	2	3	4	5
x	30	26	16	20	19
y	10	11	16	17	18



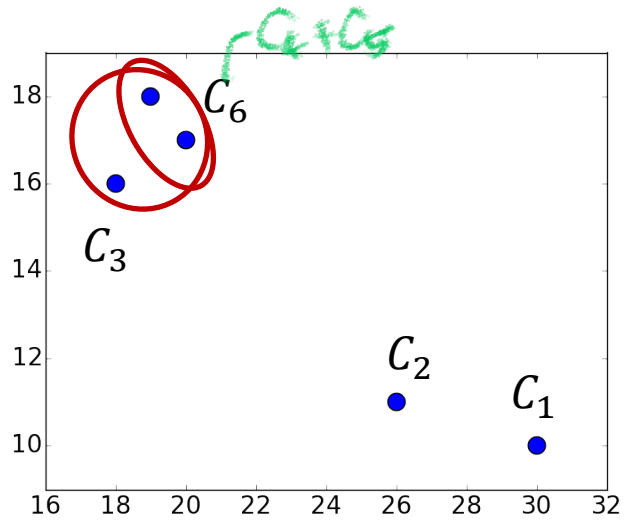
Example: Single Linkage Clustering

- Find clusters through single linkage hierarchy clustering
  - Merge cluster 4 and 5 to create new cluster

	1	2	3	4	5
x	30	26	16	20	19
y	10	11	16	17	18

Distance $d(A, B) = \min\{d(a, b): a \in A, b \in B\}$

	1	2	3	6
1	0			
2	4.12	0		
3	15.23	11.18	0	
6	12.21	8.48	**3.61**	0



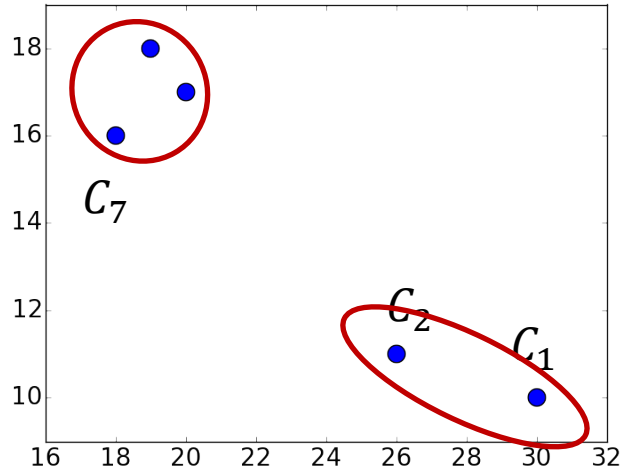
Example: Single Linkage Clustering

- Find clusters through single linkage hierarchy clustering
  - Merge cluster 3 and 6 to create new cluster

	1	2	3	4	5
x	30	26	16	20	19
y	10	11	16	17	18

Distance $d(A, B) = \min\{d(a, b): a \in A, b \in B\}$

	1	2	7
1	0		
2	**4.12**	0	
7	12.21	8.48	0



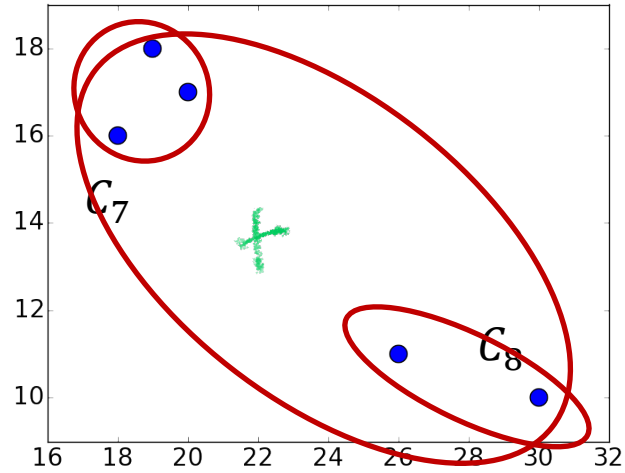
Example: Single Linkage Clustering

- Find clusters through single linkage hierarchy clustering
  - Merge cluster 1 and 2 to create new cluster

	1	2	3	4	5
x	30	26	16	20	19
y	10	11	16	17	18

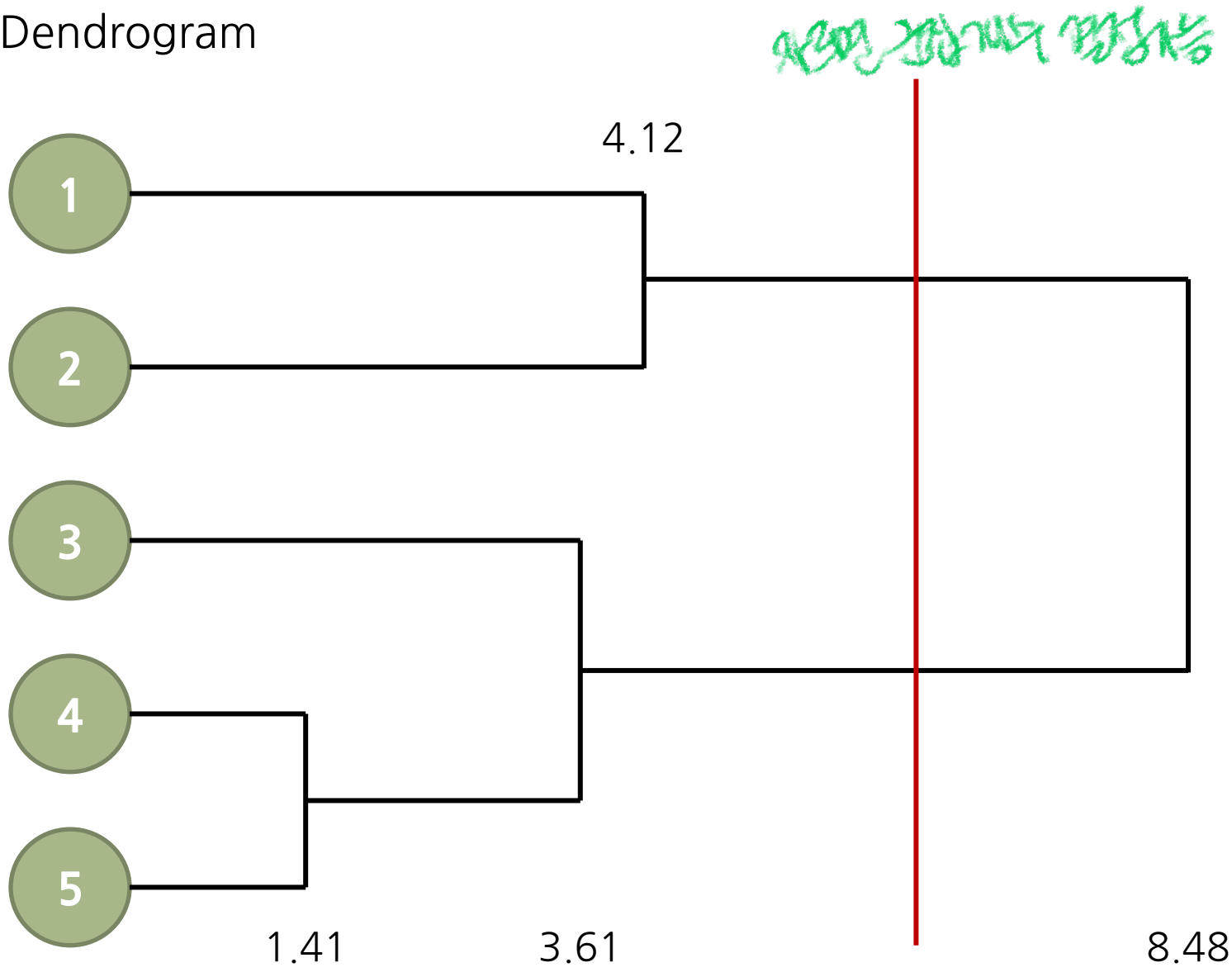
Distance $d(A, B) = \min\{d(a, b): a \in A, b \in B\}$

	7	8
7	0	
8	8.48	0



Example: Single Linkage Clustering

- Dendrogram



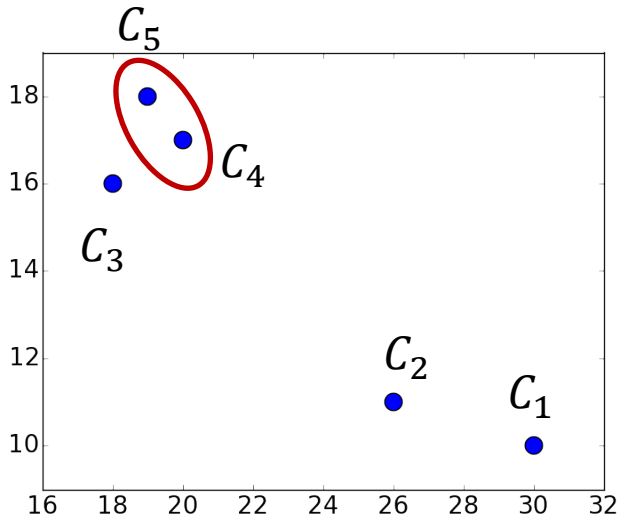
Example: Complete Linkage Clustering

- Find clusters through complete linkage hierarchy clustering
  - Start each data as own cluster

	1	2	3	4	5
x	30	26	16	20	19
y	10	11	16	17	18

Distance $d(A, B) = \max\{d(a, b): a \in A, b \in B\}$

	1	2	3	4	5
1	0				
2	4.12	0			
3	15.23	11.18	0		
4	12.21	8.48	4.12	0	
5	13.60	9.90	3.61	**1.41**	0



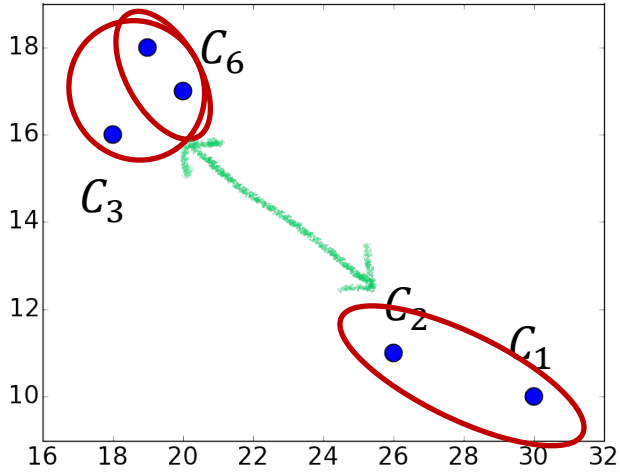
Example: Complete Linkage Clustering

- Find clusters through complete linkage hierarchy clustering
  - Merge cluster 4 and 5 to create new cluster

	1	2	3	4	5
x	30	26	16	20	19
y	10	11	16	17	18

Distance $d(A, B) = \max\{d(a, b): a \in A, b \in B\}$

	1	2	3	6
1	0			
2	**4.12**	0		
3	15.23	11.18	0	
6	13.60	9.90	**4.12**	0



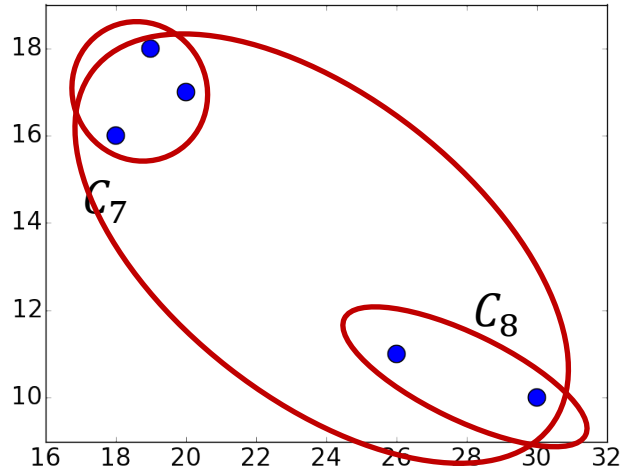
Example: Complete Linkage Clustering

- Find clusters through complete linkage hierarchy clustering
  - Merge cluster 1 and 2 to create new cluster
  - Merge cluster 3 and 6 to create new cluster

	1	2	3	4	5
x	30	26	16	20	19
y	10	11	16	17	18

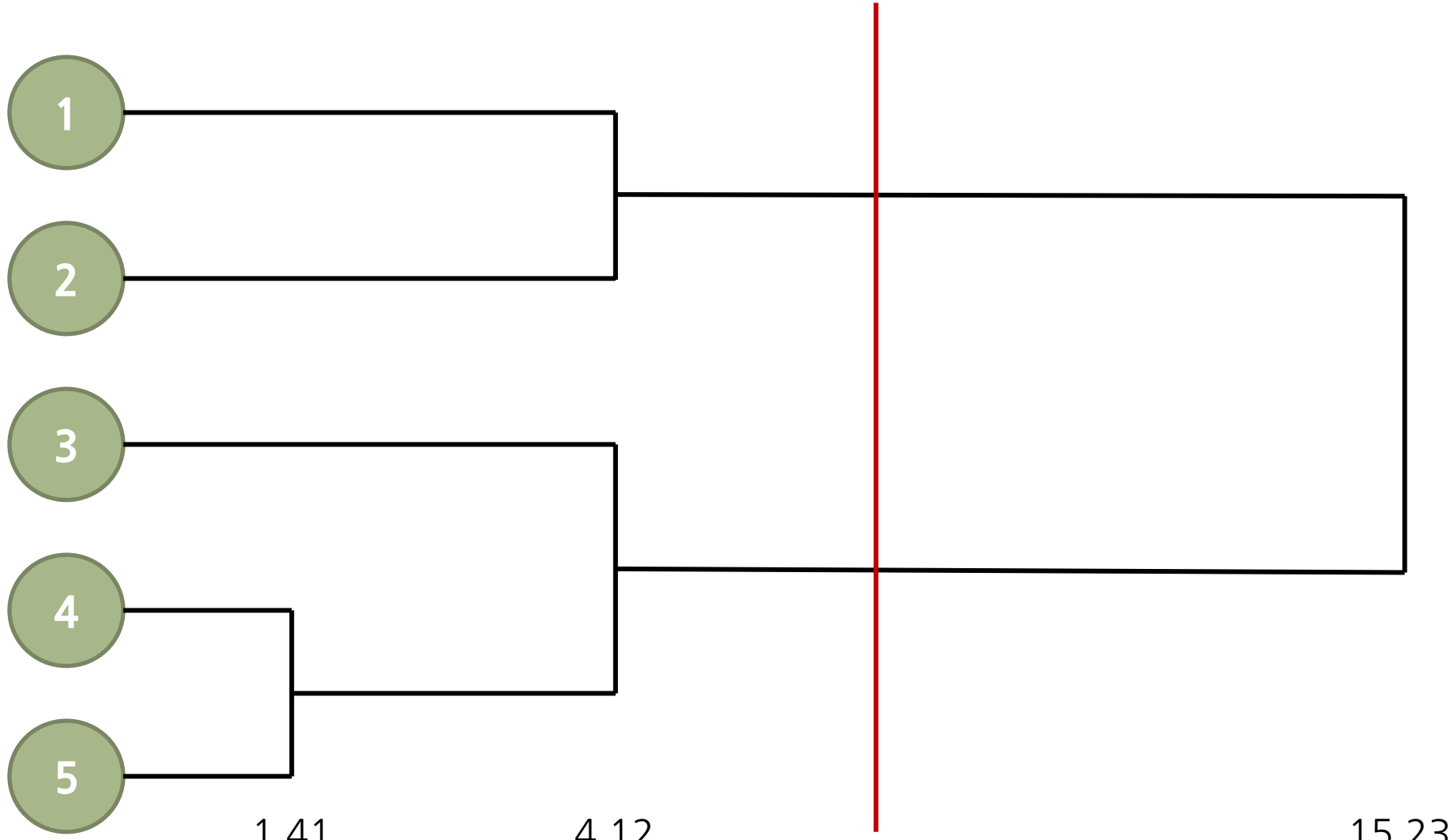
Distance $d(A, B) = \max\{d(a, b): a \in A, b \in B\}$

	7	8
7	0	
8	15.23	0



Example: Complete Linkage Clustering

- Dendrogram



Divisive Clustering - DIANA

- Divisive method starts with one cluster including all samples
  - At each step, divide cluster into two sub clusters until every cluster consists of one data point
    각 단계에서 클러스터를 두개의 하위 클러스터로 나누며, 최종 적으로는 각 클러스터가 하나만의 데이터만 가지도록 한다.
  - This algorithm is based on the average distance between one object and the others
    이 알고리즘은 한 데이터 포인트와 다른 포인트들 가의 평균거리를 기준으로 함.

$$\bar{d}(i,C) = \begin{cases} \dfrac{1}{|C|-1}\displaystyle\sum_{j\in C, j\neq i} d(i,j), & if\ i \in C \\ \dfrac{1}{|C|}\displaystyle\sum_{j\in C} d(i,j), & if\ i \notin C \end{cases}$$

  - i represent i-th object

DIANA Algorithm

1. • Consider all samples as one cluster
   모든 샘플을 하나의 클러스터로 간주

2. • Select the cluster C containing two objects with the longest distance
   가장 멀리 떨어진 두개의 객체를 포함하는 클러스터 C 생성

3. **클러스터 C를 두 부분으로 나눈다**
   • Divide cluster C into two as follows (At first, C' is empty set(ϕ))
   - Find object i with maximum $\bar{d}(i, C)$ **C내에서 다른 점들과의 평균 거리가 가장 먼 객체 i를 찾는다**
   - $C \leftarrow C - \{i\}, C' \leftarrow C' \cup \{i\}$
   - If there exist the objects j in C whose $e(j) = \bar{d}(j, C) - \bar{d}(j, C') > 0$, select one of them with maximum $e(j)$, remove j from C and add j into C'
   - If $e(j) < 0$ for all objects in C, finish this step
     C에 남아있는 도는 객체의 e(j)가 음수이면 분할 종료, 더이상 옮길만한 데이터 없음

4. • Repeat step 2 and 3 until the number of clusters is the same as the number of samples
   클러스터 수가 데이터 수와 같아질 때까지 2,3단계 반복

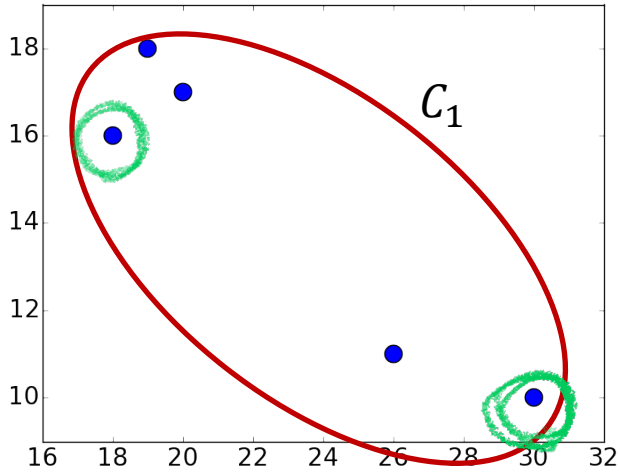
Example: DIANA

- Find clusters through DIANA
  - Start with a cluster consisting of all objects
  - $C_1 = \{1,2,3,4,5\}$
  - $C_2 = \{\}$

	1	2	3	4	5
x	30	26	16	20	19
y	10	11	16	17	18

Step 2: Find pair of objects wit the longest distance

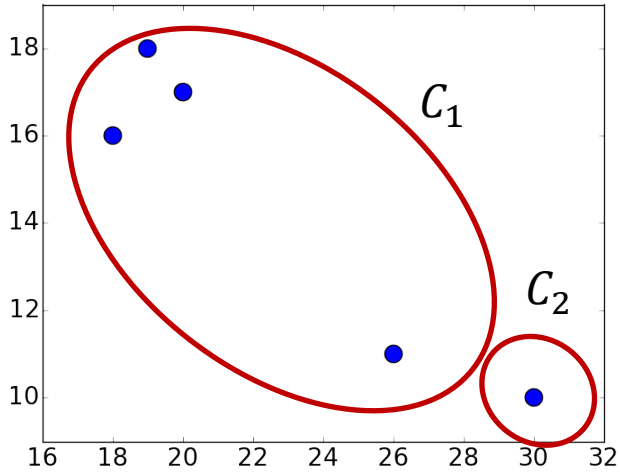
$d(i,j)$	1	2	3	4	5
1	0				
2	4.12	0			
3	**15.23**	11.18	0		
4	12.21	8.48	4.12	0	
5	13.60	9.90	3.61	1.41	0



Example: DIANA

- Find clusters through DIANA
  - C_1 is the selected cluster

	1	2	3	4	5
x	30	26	16	20	19
y	10	11	16	17	18



Step 3

$d(i,j)$	1	2	3	4	5
1	0	4.12	15.23	12.21	13.60
2	4.12	0	11.18	8.48	9.90
3	15.23	11.18	0	4.12	3.61
4	12.21	8.48	4.12	0	1.41
5	13.60	9.90	3.61	1.41	0

↓ Average except 0

	1	2	3	4	5
$\bar{d}(i, C_1)$	**11.29**	8.42	8.54	6.56	7.13

계산방법

$$\bar{d}(i,C) = \begin{cases} \frac{1}{|C|-1}\sum_{j \in C, j \neq i} d(i,j) & i \in C \\ \frac{1}{|C|}\sum_{j \in C} d(i,j) & i \notin C \end{cases}$$

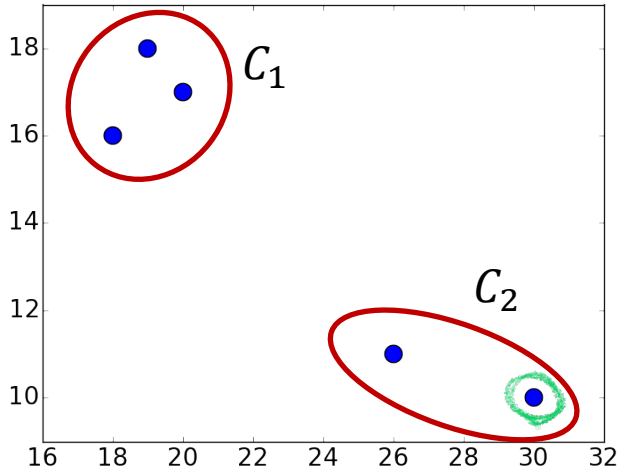
Example: DIANA

- Find clusters through DIANA
  - $C_1 = \{2,3,4,5\}, C_2 = \{1\}$

	1	2	3	4	5
x	30	26	16	20	19
y	10	11	16	17	18

Step 3

	2	3	4	5
$\bar{d}\,(i, C_1)$	9.85	6.30	4.67	4.97
$\bar{d}\,(i, C_2)$	4.12	15.2	12.2	13.6
$e(i)$	**5.73**	-8.9	-7.53	-8.63



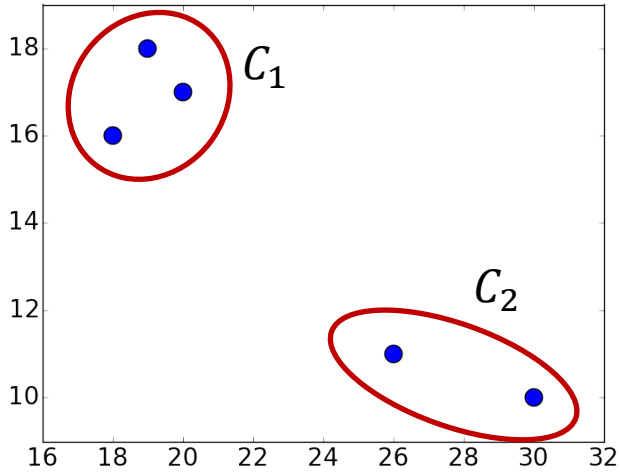
Example: DIANA

- Find clusters through DIANA
  - $C_1 = \{3,4,5\}$, $C_2 = \{1,2\}$

	1	2	3	4	5
x	30	26	16	20	19
y	10	11	16	17	18

Step 3

	3	4	5
$\bar{d}\,(i, C_1)$	3.87	2.77	2.51
$\bar{d}\,(i, C_2)$	13.21	10.35	11.75
$e(i)$	-9.34	-7.58	-9.24



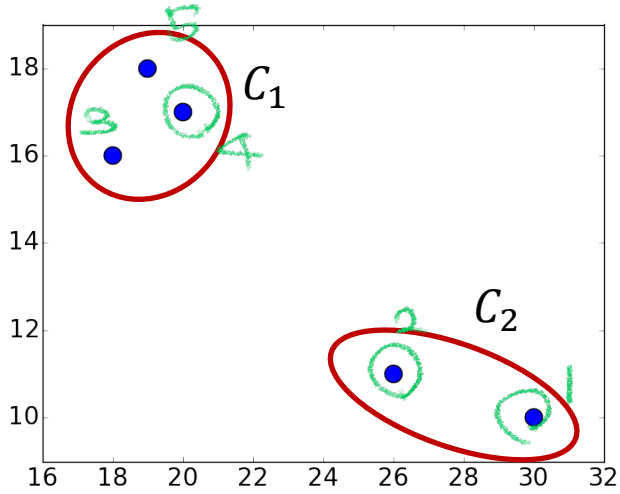
Example: DIANA

- Find clusters through DIANA
  - $C_1 = \{3,4,5\}, C_2 = \{1,2\}$
  - Find pair of objects wit the longest distance

	1	2	3	4	5
x	30	26	16	20	19
y	10	11	16	17	18

Step 2: Find pair of objects wit the longest distance

$d(i,j)$	1	2	3	4	5
1	0				
2	**4.12**	0			
3			0		
4			**4.12**	0	
5			3.61	1.41	0



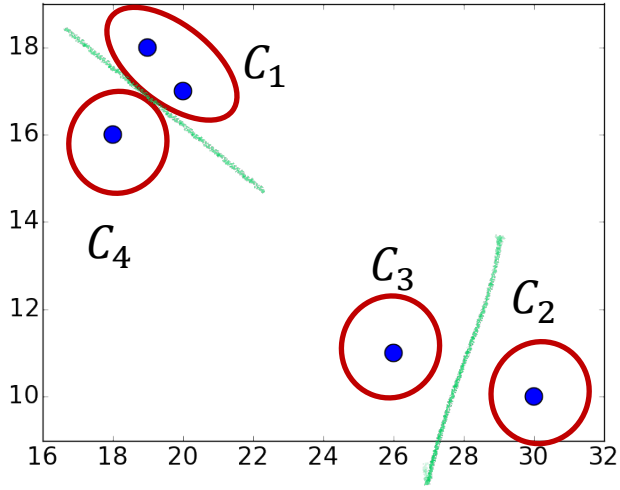
Example: DIANA

- Find clusters through DIANA
  - $C_1 = \{4,5\}, C_4 = \{3\}$

	1	2	3	4	5
x	30	26	16	20	19
y	10	11	16	17	18

Step 3

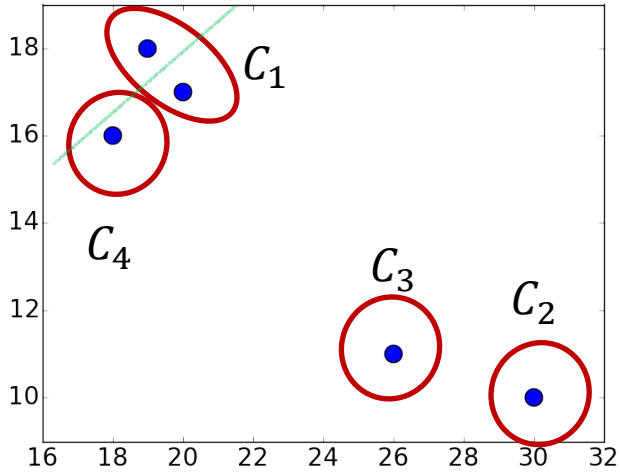
	4	5
$\bar{d}\ (i, C_1)$	1.41	1.41
$\bar{d}\ (i, C_4)$	4.12	3.61
$e(i)$	-2.71	-2.20



Example: DIANA

- Find clusters through DIANA
  - Select C_2
  - $C_1 = \{1,2\}, C_3 = \{\}$
  - C_2 contains only two object, so divide C_2 into two clusters directly: $C_2 = \{1\}, C_3 = \{2\}$
  - Select C_1
  - $C_1 = \{3,4,5\}, C_4 = \{\}$

	1	2	3	4	5
x	30	26	16	20	19
y	10	11	16	17	18



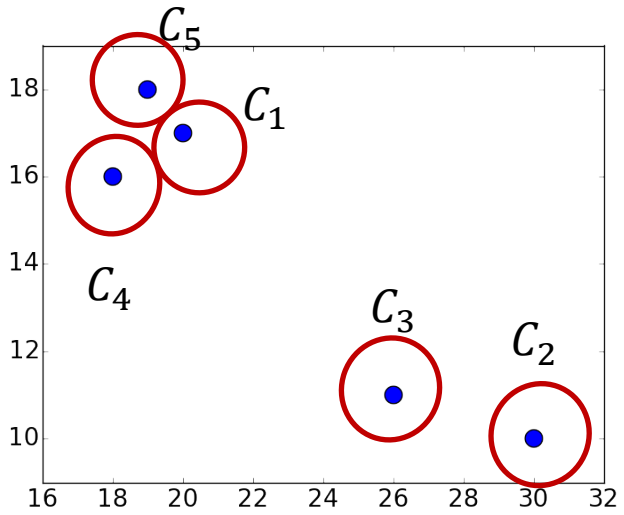
Step 3

	3	4	5
$\bar{d}(i, C_1)$	**3.87**	2.77	2.51

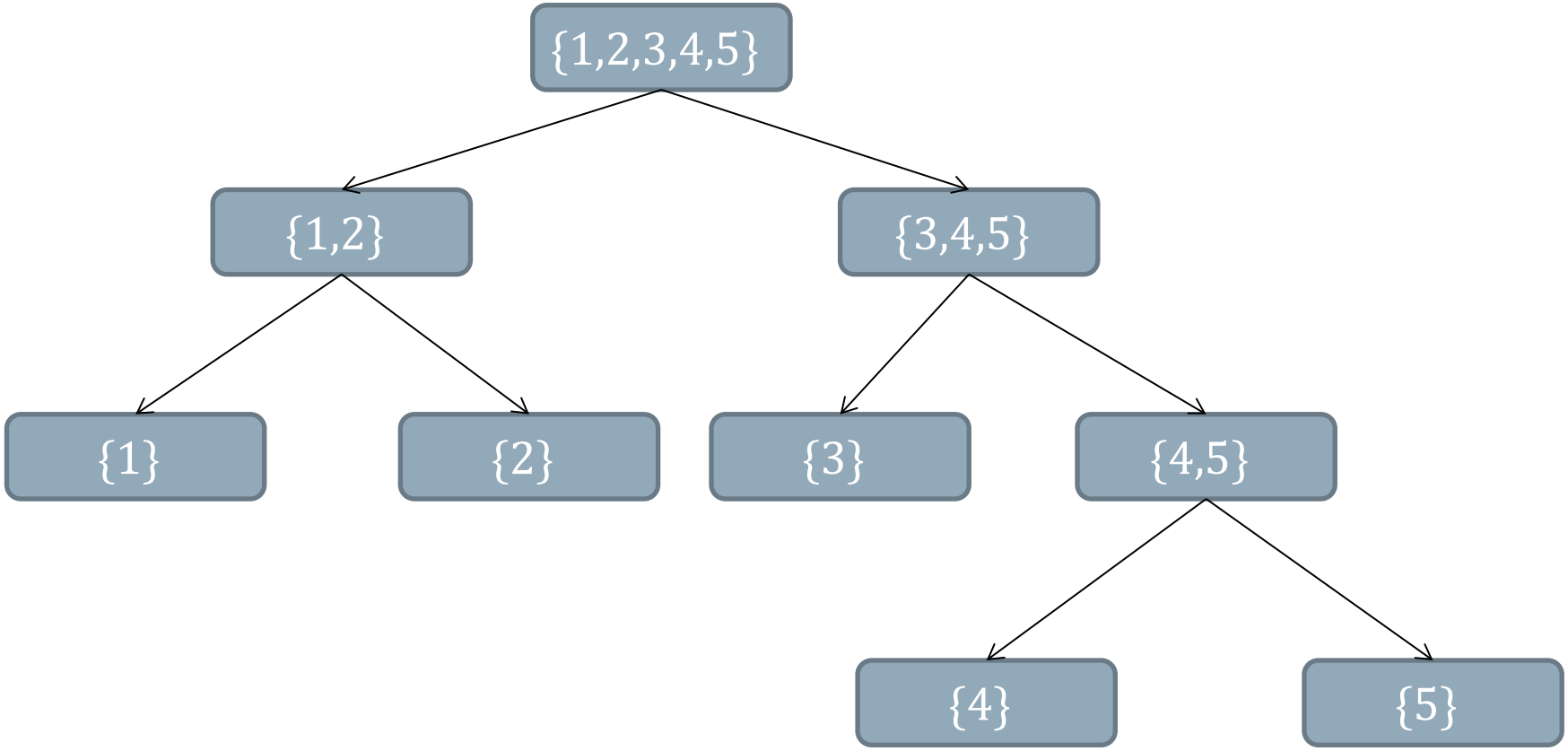
Example: DIANA

- Find clusters through DIANA
  - Select C_1
  - $C_1 = \{4,5\}, C_5 = \{\}$
  - C_1 contains only two object, so divide C_1 into two clusters directly: $C_1 = \{4\}, C_5 = \{5\}$

	1	2	3	4	5
x	30	26	16	20	19
y	10	11	16	17	18

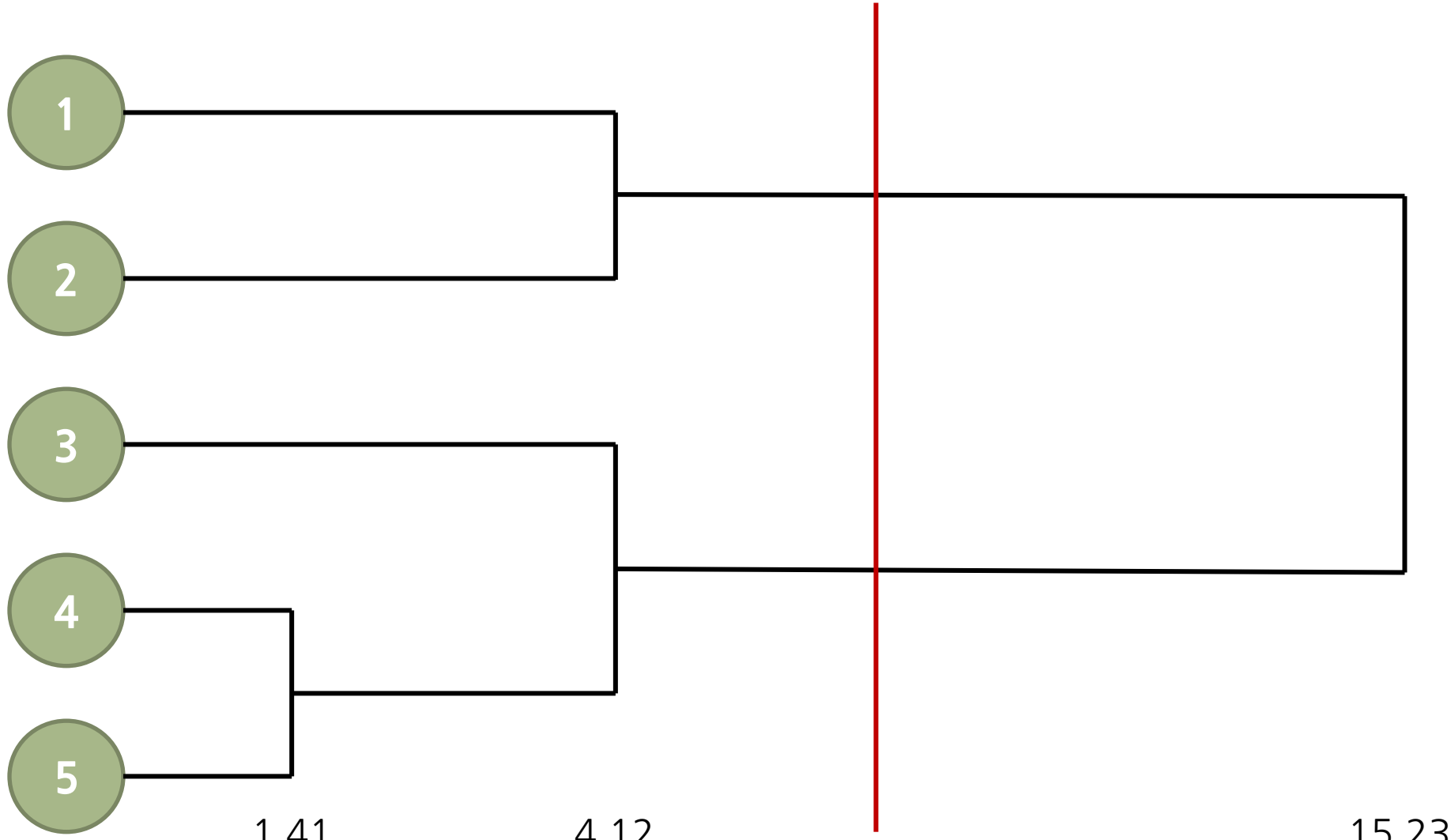


Example: DIANA



Example: DIANA

- Dendrogram



Hierarchical Clustering: Pros and Cons

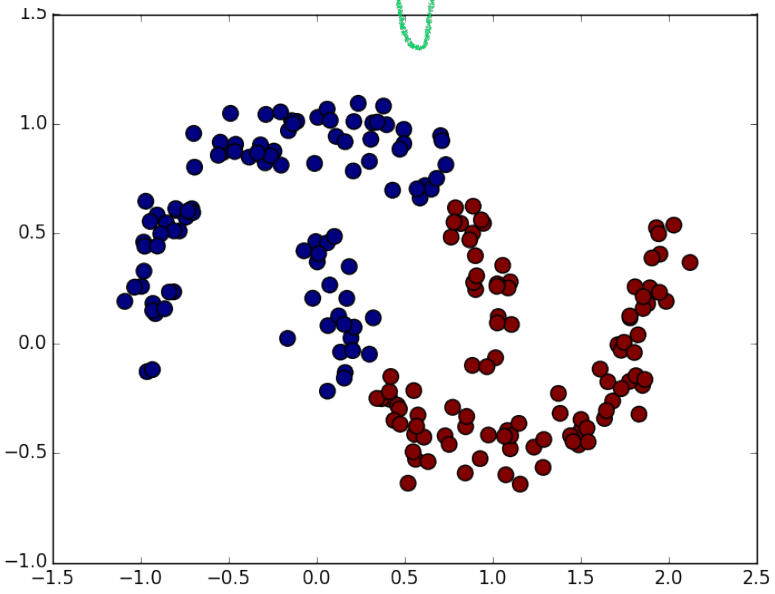
- Pros
  - No need to specify k in advance
  - Produces a hierarchical structure
- Cons
  - Computationally expensive for large datasets
  - Sensitive to noise

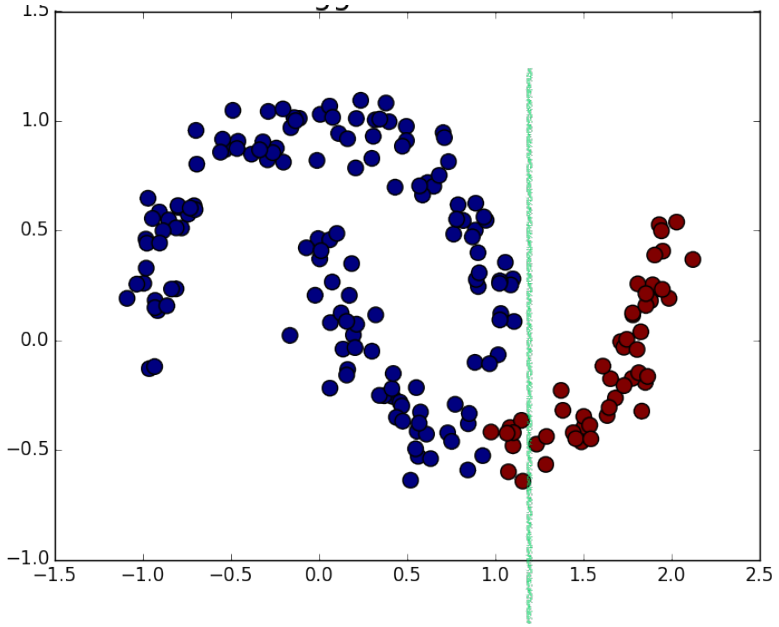
Evaluation Measures

How to Measure Clustering Quality

- Clustering problem is unsupervised problem
  - No explicit answer for learning
  - We need to define a method to measure quality of clustering

Which one is better?





How to Measure Clustering Quality

- Measures that do not require ground truth labels
  - Inertia
    - Within-cluster sum-of-squares

$$\sum_{i=0}^{n} \min_{\mu_j \in C} \|x_j - \mu_i\|^2$$

  - Silhouette Coefficient
    - $s(i)$: Silhouette coefficient of i-th sample
    - $a(i)$: The mean distance between a sample and all other points in the same class
    - $b(i)$: The mean distance between a sample and all other points in the next nearest cluster

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

$$-1 \le s(i) \le 1$$

    - Overall clustering quality can be obtained by averaging $s(i)$ for all samples

How to Measure Clustering Quality

- Clustering performance evaluation measure
  - Homogeneity: each cluster contains only members of a single class

$$h = 1 - \frac{H(C|K)}{H(C)}$$

    - $H(C)$ is the entropy of the classes

$$H(C) = -\sum_{c=1}^{|C|} \frac{n_c}{n} \cdot \log\left(\frac{n_c}{n}\right)$$

    - $H(C|K)$ is the conditional entropy of the classes given the cluster assignments

$$H(C|K) = -\sum_{c=1}^{|C|}\sum_{k=1}^{|K|} \frac{n_{c,k}}{n} \log\left(\frac{n_{c,k}}{n_k}\right)$$

    - n is the total number of samples, n_c and n_k are the number of samples respectively belonging to class c and cluster k
    - $n_{c,k}$ is the number of samples from class c assigned to cluster k
  - Completeness: all members of a given class are assigned to the same cluster

$$c = 1 - \frac{H(K|C)}{H(K)}$$

How to Measure Clustering Quality

- Clustering performance evaluation measure
  - Adjusted Rand Index(ARI)
    - Given the knowledge of the ground truth class assignments and our clustering algorithm assignments of the same samples, the adjusted Rand index is a function that measures the similarity of the two assignments

$$\mathrm{ARI} = \frac{\mathrm{RI} - \mathbb{E}[\mathrm{RI}]}{\max(\mathrm{RI}) - \mathbb{E}[\mathrm{RI}]}$$

    - C is a ground truth class assignment, K is the clustering
    - a is the number of pairs of elements that are in the same set in C and in the same set in K
    - b is the number of pairs of elements that are in different sets in C and in different sets in K
    - Raw Rand index $\mathrm{RI} = \frac{a+b}{C_2^n}$($C_2^n$ is the total number of possible pairs in the dataset)

$$C_2^n = \frac{n!}{2!\,(n-2)!}$$

How to Measure Clustering Quality

- Contingency table

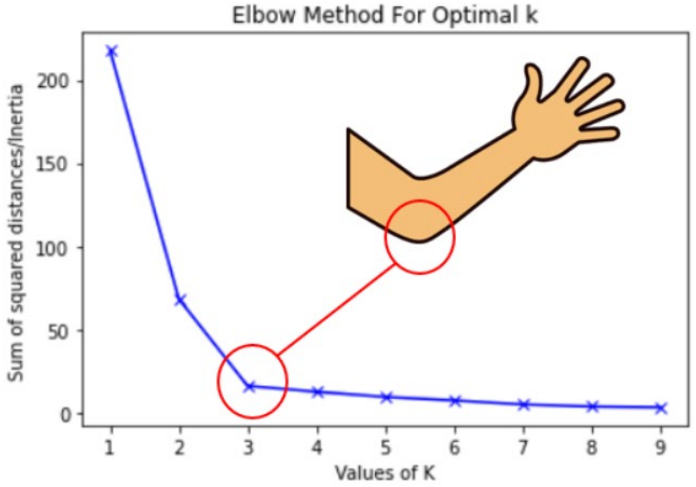
	K_1	K_2	$\cdots$	K_s	$sums$
C_1	n_{11}	n_{12}	$\cdots$	n_{1s}	a_1
C_2	n_{21}	n_{22}	$\cdots$	n_{2s}	a_2
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
C_r	n_{r1}	n_{r2}	$\cdots$	n_{rs}	a_r
$sums$	b_1	b_2	$\cdots$	b_s	n

$$\mathrm{ARI} = \frac{\mathrm{RI} - \mathbb{E}[\mathrm{RI}]}{\max(\mathrm{RI}) - \mathbb{E}[\mathrm{RI}]} = \frac{\sum_{ij}\binom{n_{ij}}{2} - \left[\sum_i \binom{a_i}{2}\sum_j \binom{b_j}{2}\right] / \binom{n}{2}}{\frac{1}{2}\left[\sum_i \binom{a_i}{2} + \sum_j \binom{b_j}{2}\right] - \left[\sum_i \binom{a_i}{2}\sum_j \binom{b_j}{2}\right] / \binom{n}{2}}$$

$$\binom{n}{k} = {}_nC_k$$

Determining the Optimal Number of Clusters

- Elbow method
  - Plot the total within-cluster sum of squares against the number of clusters
  - Look for an “elbow” point where the rate of decrease sharply slows



Line plot between K and inertia

- Silhouette Score
  - Measures how similar an object is to its own cluster compared to other clusters
  - Value ranges from -1 to 1 and a higher value indicates better clustering