머신러닝응용 제08강

Clustering Analysis

첨단공학부 김동하교수

제08강 Clustering Analysis

1	거리 혹은 비유사도의 개념에 대해 학습한다.
2	계층적 군집분석에 대해 학습한다.
3	K-군집 분석법에 대해 학습한다.

핵심 단어

- 최장.최단 연결법
- 덴드로그램
- K-평균 군집법

08강. Clustering Analysis

01. 군집분석이란

1) 군집 분석의 정의

- 군집분석은 모집단 또는 범주에 대한 사전 정보가 없는 경우에 사용 -> 비지도 학습법

- 주어진 관측값들 사이의 거리 또는 유사성을 이용하여 전체를 몇 개의 집단으로 그룹화

- 각 집단의 성격을 파악함으로써 데이터 전체의 구조에 대한 이해를 돕고자 하는 것이 목표

2) 군집화

군집화의 기준

- 동일한 군집에 속하는 개체는 비슷하고, 다른 군집에 속한 개체는 그렇지 않도록 군집을 구성

군집화를 위한 변수

- 예: 고객 세분화
- 인구통계적 변수 (성별, 나이, 거주지, 소득 등)
- 구매패턴 변수 (구매상품, 주기, 거래액 등)

3) 군집 분석의 활용

고객 세분화

- 고객이 기업의 수익에 기여하는 정도를 통한 고객 세분화
- 개별 고객에 대한 맞춤 관리

고객의 구매패턴에 따른 고객 세분화

- 신상품 판촉, 교차판매를 위한 표적 집단 구성

4) 군집 분석의 유형

- 상호 배반적 (disjoint) 군집
  - 각 관측치가 오직 하나의 군집에만 속함
    - 예: 한국인, 중국인, 일본인, …

- 계층적 (hierarchical) 군집
  - 한 군집이 다른 군집의 내부에 포함
  - 군집간의 중복은 없으며, 군집들이 매 단계 계층적인 구조를 이룸.
    - 예: 전자제품 -> 주방용 -> 냉장고

4) 군집 분석의 유형

- 중복 (overlapping) 군집
  - 두 개 이상의 군집에 한 관측치가 동시에 포함.

4) 군집 분석의 특징

- 군집 분석을 위한 기준의 설정에 있어서 다양한 방법이 존재.
  - 유사성 혹은 비유사성의 다양한 정의

- 분석자의 주관에 따라 결과가 달라질 수 있음.

02. 거리

1) 비유사성의 척도: 거리

- 관측값들이 서로 얼마나 유사한지 측정할 수 있는 측도가 필요

- 군집분석에서는 비유사성을 기준으로 하여 유사정도를 측정.

2) 거리 측도의 종류들

- 유클리디안 거리 (Euclidean distance)
  - p차원 공간에서 주어진 두 점 $x = (x_1, x_2, \cdots, x_p)$와 $y = (y_1, y_2, \cdots, y_p)$사이의 유클리디안 거리는 다음과 같이 정의된다:

$$d(x, y) = \left[\sum_{i=1}^{p} (x_i - y_i)^2 \right]^{1/2}.$$

2) 거리 측도의 종류들

- Minkowski 거리

$$d(x, y) = \left[\sum_{i=1}^{p} (x_i - y_i)^{\mathbf{q}}\right]^{1/\mathbf{q}} .$$

- 표준화 거리

$$d(x, y) = \left[\sum_{i=1}^{p} \left(\frac{x_i - y_i}{S_i}\right)^2\right]^{1/2} .$$

3) 범주형 자료의 거리

- 불일치 항목 수

- 예:

개체	성별	학력	출신지역
A	남자	고졸	경기
B	여자	고졸	전남
C	남자	대졸	경기

- $d(A,B)=2,\ d(A,C)=1,\ d(B,C)=3$

08강. Clustering Analysis

03. 계층적 군집 분석

1) 계층적 군집 분석

- 가까운 관측값들 끼리 묶는 병합 방법과 먼 관측값들을 나누어가는 분할 방법으로 나눌 수 있다.

- 계층적 군집분석에서는 주로 병합 방법이 주로 사용됨.

- 나무 구조인 덴드로그램을 통해 나타낼 수 있음.

2) 병합 방법

- 처음에 n개의 자료를 각각 하나의 군집으로 생각한다 (n개의 군집).
- 가장 거리가 가까운 두 개의 군집을 병합하여 n-1개의 군집으로 군집을 줄인다.
- 그 다음, n-1개의 군집 중 가장 가까운 두 군집을 병합하여 군집을 n-2개로 줄인다.

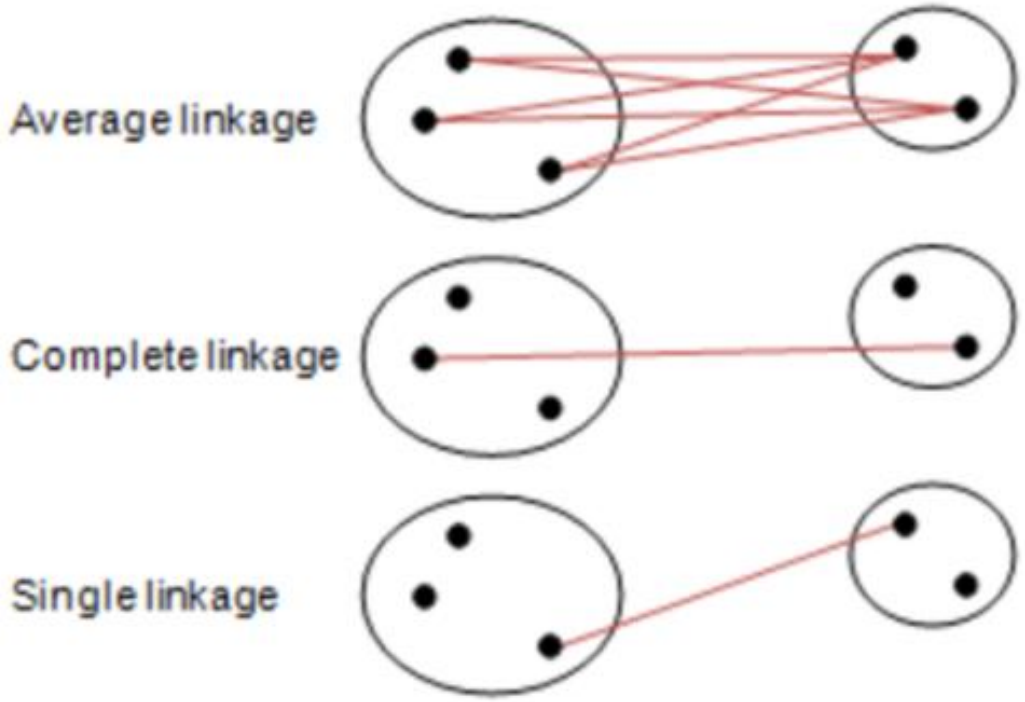
2) 병합 방법

- 이를 반복하여 계속하여 군집의 수를 줄여 나간다.

- 이 과정은 시작부분에는 군집의 크기는 작고 동질적이며, 끝부분에서는 군집의 크기는 커지고 이질적이 된다.

2) 병합 방법

- 군집들간의 거리를 측정하는 방법에 따라 다양한 종류의 방법이 있다.
  - 최단거리, 최장거리, 평균거리 방법 등.



출처: https://wikidocs.net/84773

3) 최단 연결법 (Single linkage method)

- 두 군집 사이의 거리를 각 군집에서 하나씩 관측값을 뽑았을 때 나타날 수 있는 거리의 최소값으로 측정

- 군집이 고리 형태로 연결되어 있는 경우에는 부적절한 결과를 제공.

- 고립된 군집을 찾는데 중점을 둔 방법.

3) 최단 연결법 (Single linkage method)

◆주어진 5개의 관측값에 대한 거리(비유사성) 행렬에 대하여 최단연결법으로 군집을 얻고 덴드로그램으로 나타내보자.

1	0				
2	7	0			
3	1	6	0		
4	9	3	8	0	
5	8	5	7	4	0

3) 최단 연결법 (Single linkage method)

- 1단계
  - 거리 행렬에서 d(1,3)=1이 최소이므로 관측값 1과 3을 묶어 군집 (1,3)을 만든다.
- 군집 (1,3)과 관측값 2,4,5와의 거리

$$d((2),(1,3)) = \min\{d(2,1), d(2,3)\} = d(2,3) = 6$$

$$d((4),(1,3)) = \min\{d(4,1), d(4,3)\} = d(4,3) = 8$$

$$d((5),(1,3)) = \min\{d(5,1), d(5,3)\} = d(5,3) = 7$$

- 이를 이용하여 다음 장과 같은 거리 행렬을 만든다.

3) 최단 연결법 (Single linkage method)

2단계

- d(2,4)=3이 최소이므로 관측값 2와 4를 묶어 군집 (2,4)를 만들자.

(1,3)	0			
2	6	0		
4	8	3	0	
5	7	5	4	0

3) 최단 연결법 (Single linkage method)

2단계

- 군집 (2,4), (1,3), (5)사이의 거리행렬 구하기.

$$d((2,4),(1,3)) = \min\{d((2),(1,3)), d((4),(1,3))\} = d((2),(1,3)) = 6$$

$$d((5),(2,4)) = \min\{d(5,2), d(5,4)\} = d(5,4) = 4$$

(1,3)	0		
(2,4)	6	0	
5	7	4	0

3) 최단 연결법 (Single linkage method)

- 3단계
  - d((5),(2,4))=4이 최소값을 가지므로 군집 (2,4)와 (5)를 묶어 (2,4,5)를 만든다.
  - d((1,3),(2,4,5))=d(2,3)=6을 이용하여 다음의 거리 행렬을 얻는다.

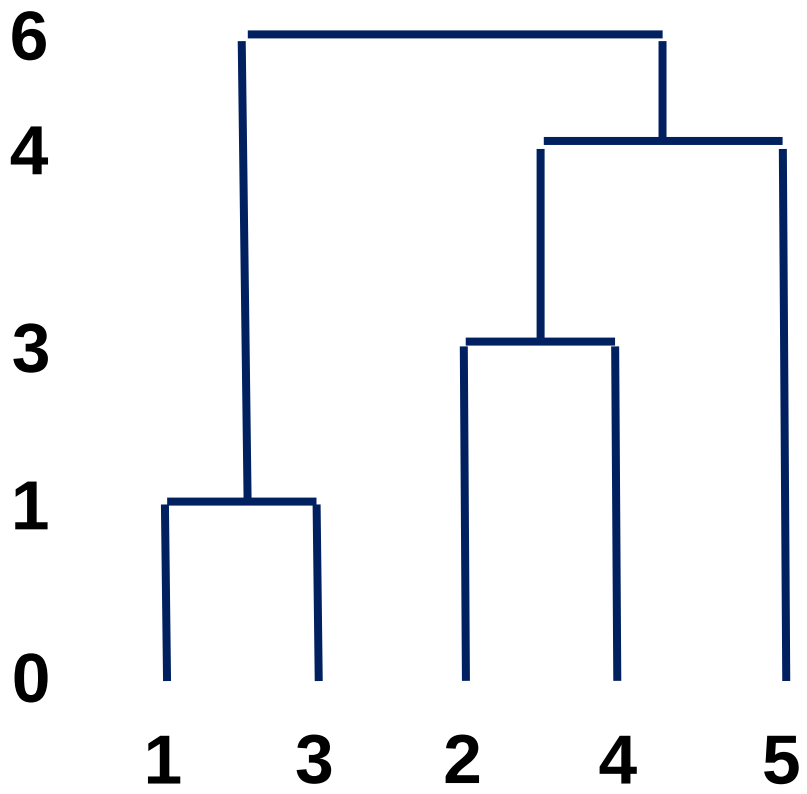
(1,3)	0	
(2,4,5)	6	0

3) 최단 연결법 (Single linkage method)

- 4단계
  - 마지막으로 전체가 하나의 군집을 이룬다.

3) 최단 연결법 (Single linkage method)

덴드로그램 결과



4) 최장(완전) 연결법 (Complete Linkage Method)

- 두 군집 사이의 거리를 각 군집에서 하나씩 관측값을 뽑았을 때 나타날 수 있는 거리의 최대값으로 측정.
- 같은 군집에 속하는 관측치는 알려진 최대 거리보다 짧다.
- 군집들의 내부 응집성에 중점을 둔다.

08강. Clustering Analysis

04. 비계층적 군집 분석

1) 비계층적 군집분석

- 비계층적 군집분석에는 흔히 관측값들을 몇 개의 군집으로 나누기 위하여 주어진 판정기준을 최적화 한다.

- 따라서, 최적분리 군집분석이라고도 한다.

- 대표적인 비계층적 군집분석 방법
  - K-평균 방법.

2) K-평균 군집법

- 사전에 결정된 군집수 K에 기초하여 전체 데이터를 상대적으로 유사한 K개의 군집으로 구분.

- 계층적 군집법에 비하여 적은 계산량.
  - 대용량 데이터를 빠르게 처리할 수 있음.

2) K-평균 군집법

◆ 알고리즘

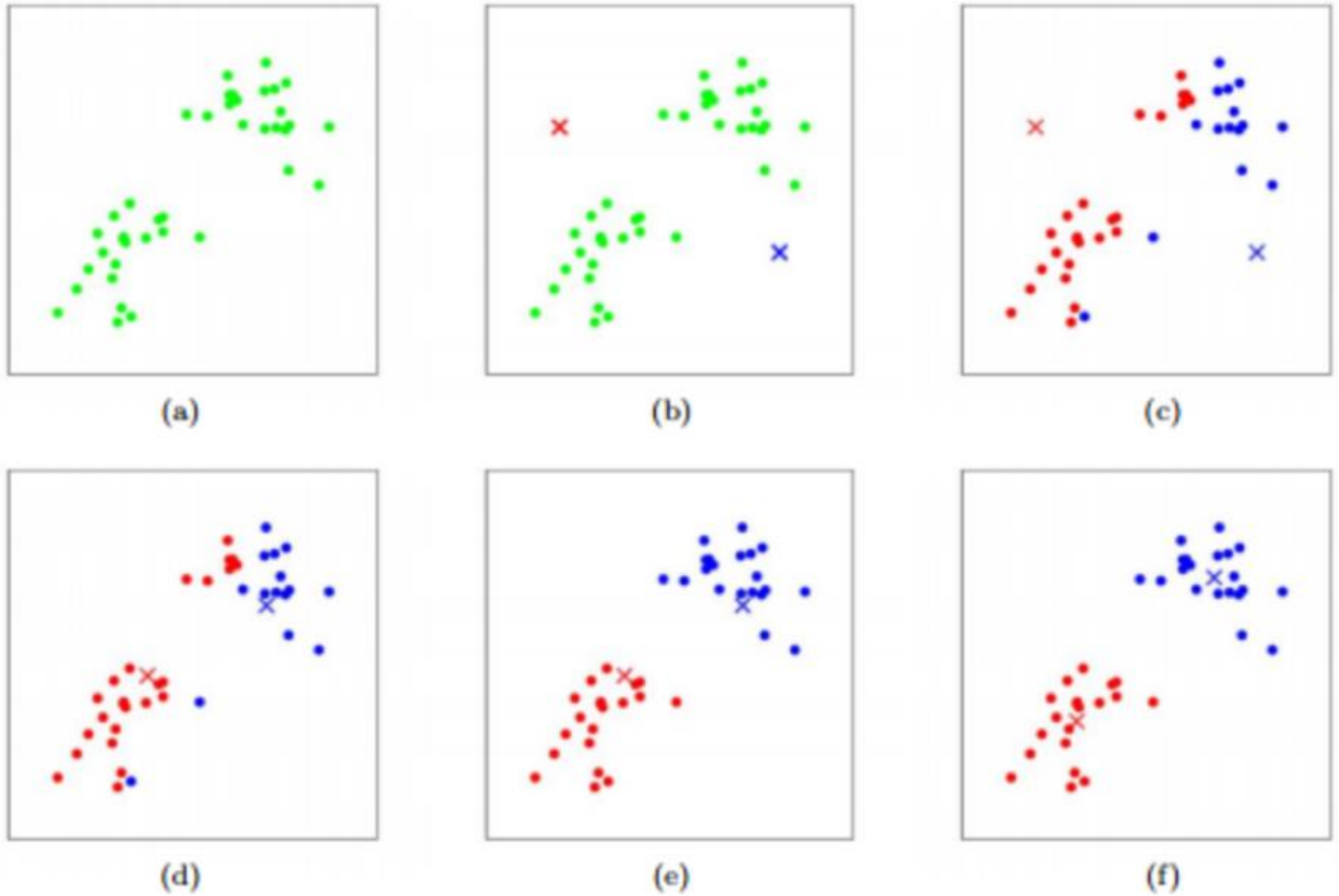
① 군집수 K를 결정한다.

② 초기 K개 군집의 중심을 선택한다.

③ 각 관측치를 그 중심과 가장 가까운 거리에 있는 군집에 할당한다.

④ 위의 과정을 기존의 중심과 새로운 중심의 차이가 없을 때까지 2-3번을 반복한다.

2) K-평균 군집법



출처: https://hleecaster.com/ml-kmeans-clustering-concept/

3) 군집 수의 결정

- K-평균 군집분석법의 결과는 군집수 K의 결정에 민감하게 반응.

- 실제 자료의 분석에서는 여러 가지의 K값에 대해 군집분석을 수행한 후 가장 좋다고 생각되는 K값을 이용.

3) 군집 수의 결정

- 일반적으로 K값이 증가함에 따라 거리제곱합은 감소.
  - 자료의 단위가 큰 영향을 미치기 때문에 표준화 작업이 먼저 수행되어야 함.

- K값을 증가시키면서 거리제곱합의 감소가 완만해지는 군집의 수를 최적의 군집 수로 결정

4) K-평균 군집분석의 단점

- 군집이 겹치는 경우에 좋지 않음.
- 이상치에 민감

08강. Clustering Analysis

05. Python을 이용한 실습

1) 데이터 설명

- Iris 데이터셋
  - 150개의 붓꽃에 대한 데이터
  - 꽃잎의 각 부분의 너비와 길이 등을 측정
  - sepal length: 꽃받침의 길이
  - sepal width: 꽃받침의 너비
  - petal length: 꽃잎의 길이
  - petal width: 꽃잎의 너비
  - species: 붓꽃의 종류 (setosa, versicolor, virginica)

1) 데이터 설명

분석 목표

- 붓꽃 데이터를 이용하여 비계층적, 계층적 군집분석 방법을 구현하자.

2) 환경설정

◆ 필요한 패키지 불러오기

```
import pandas as pd
import numpy as np
import os
import matplotlib.pyplot as plt

from sklearn.preprocessing import StandardScaler
from sklearn.cluster import KMeans
from scipy.cluster.hierarchy import linkage, dendrogram, fcluster, single, complete

from sklearn import datasets
```

3) 데이터 불러오기

◆ 50개만 랜덤하게 선택

```
iris = datasets.load_iris()
```

```
Iris = pd.DataFrame(data=iris.data, columns=iris.feature_names)
print(Iris.shape)
# featurs
# 50개만 랜덤하게 선택.
np.random.seed(1234)
random_ind = np.random.choice(np.arange(150),50)
Iris = Iris.iloc[random_ind, 0:4]
print(Iris.shape)
Iris.head()
```

```
(150, 4)
(50, 4)
```

4) 데이터 전처리

◆ 표준화하기

```
Iris_st = StandardScaler().fit_transform(Iris)
feature_names = ['sepal_length','sepal_width',
                 'petal_length','petal_width']
Iris_st = pd.DataFrame(Iris_st, columns=feature_names)
Iris_st.head()
```

	sepal_length	sepal_width	petal_length	petal_width
0	-1.385105	0.476636	-1.302324	-1.301167
1	-1.610692	0.009346	-1.357742	-1.301167
2	-0.369963	-1.626168	0.138545	0.121580
3	0.757973	0.009346	0.969816	0.768284
4	1.096353	0.476636	1.191488	1.414987

5) 계층적 군집분석-최단 연결법

유클리디안 거리 이용

```
single_dist = linkage(Iris_st,method='single',
                      metric='euclidean')
```

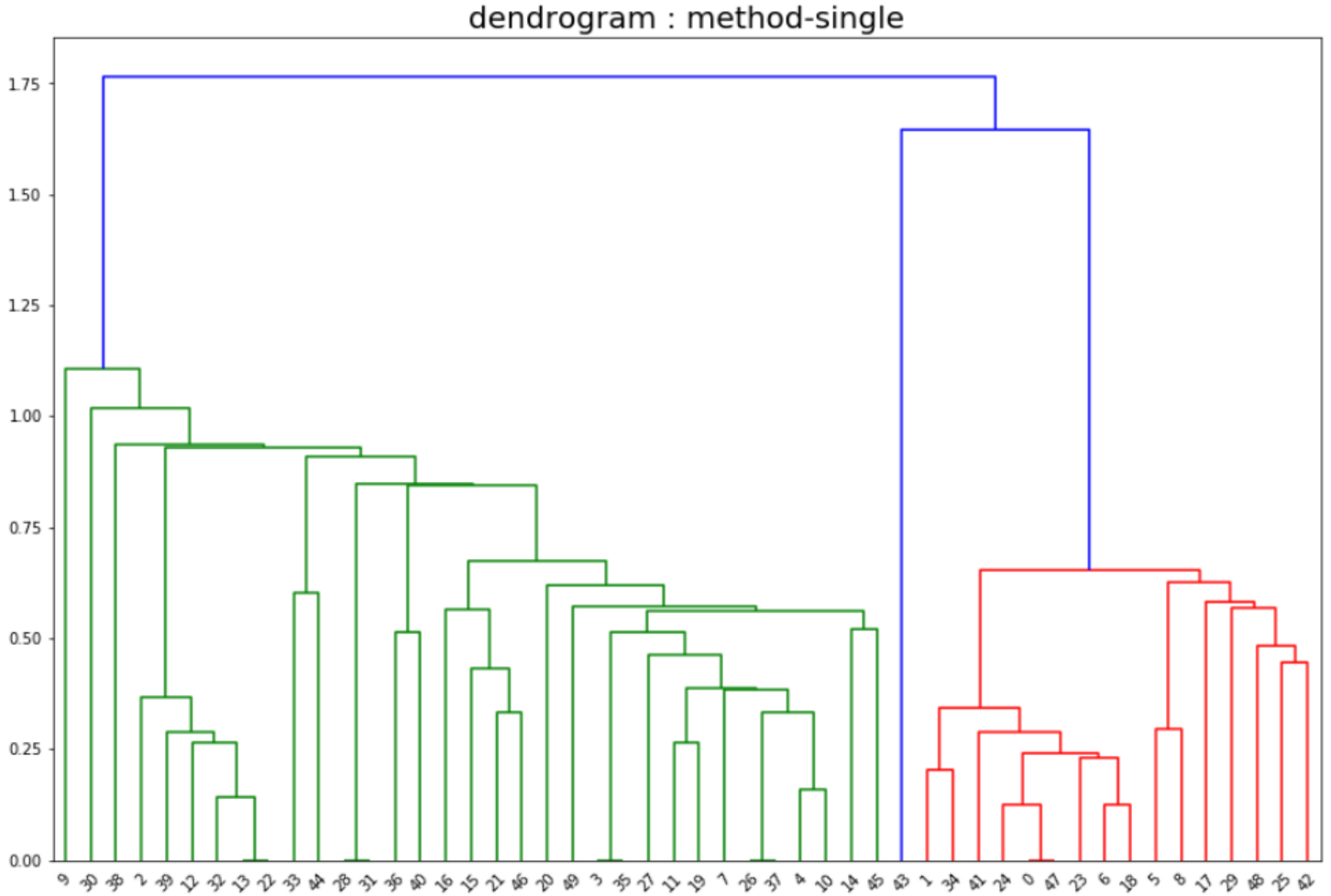
5) 계층적 군집분석-최단 연결법

덴드로그램을 이용해 결과를 표시하자.

```
plt.figure(figsize=(15,10))
dendrogram(single_dist,
           leaf_rotation=45,
           leaf_font_size=10,
)
plt.title('dendrogram : method-single', fontsize=20
plt.show()
```

5) 계층적 군집분석-최단 연결법

- 덴드로그램을 이용해 결과를 표시하자.



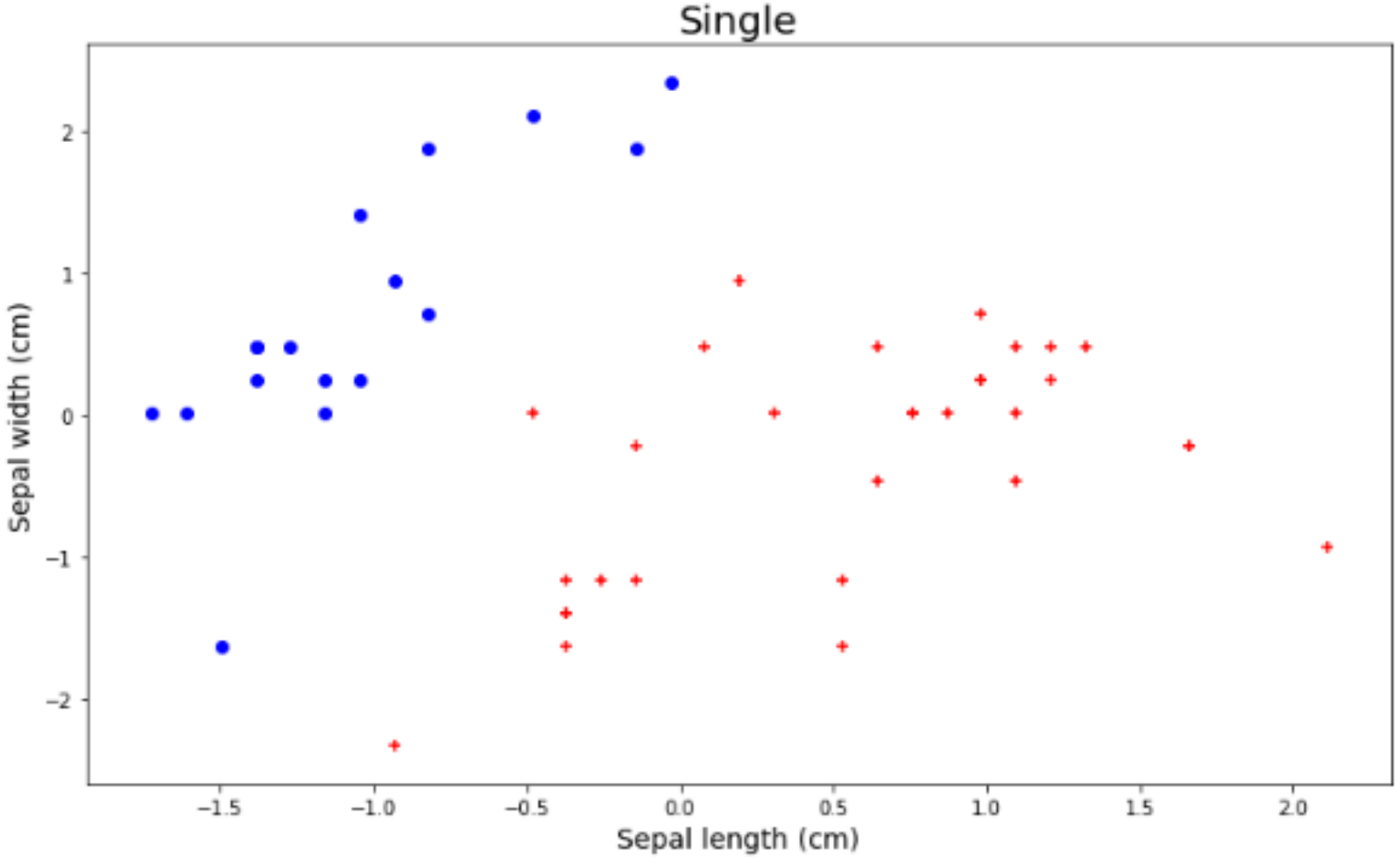
5) 계층적 군집분석-최단 연결법

◆ 군집의 수가 2일 때 분석 결과를 시각화해보자.

```
cluster_results=fcluster(single_dist,2,criterion='maxclust')
cluster_color = np.where(cluster_results == 1, 'red', 'blue')
cluster_mark = np.where(cluster_results == 1, '+', 'o')

plt.figure(1, figsize=(12, 7))
for RowIdx in range(len(Iris_st)):
    plt.scatter(Iris_st['sepal_length'].iloc[RowIdx],
                Iris_st['sepal_width'].iloc[RowIdx],
                c=cluster_color[RowIdx],
                marker=cluster_mark[RowIdx])

plt.title('Single', fontsize=20)
plt.xlabel('Sepal length (cm)', fontsize=14)
plt.ylabel('Sepal width (cm)', fontsize=14)
plt.show()
```

5) 계층적 군집분석-최단 연결법

- 군집의 수가 2일 때 분석 결과를 시각화해보자.



6) 비계층적 군집분석-K평균 군집법

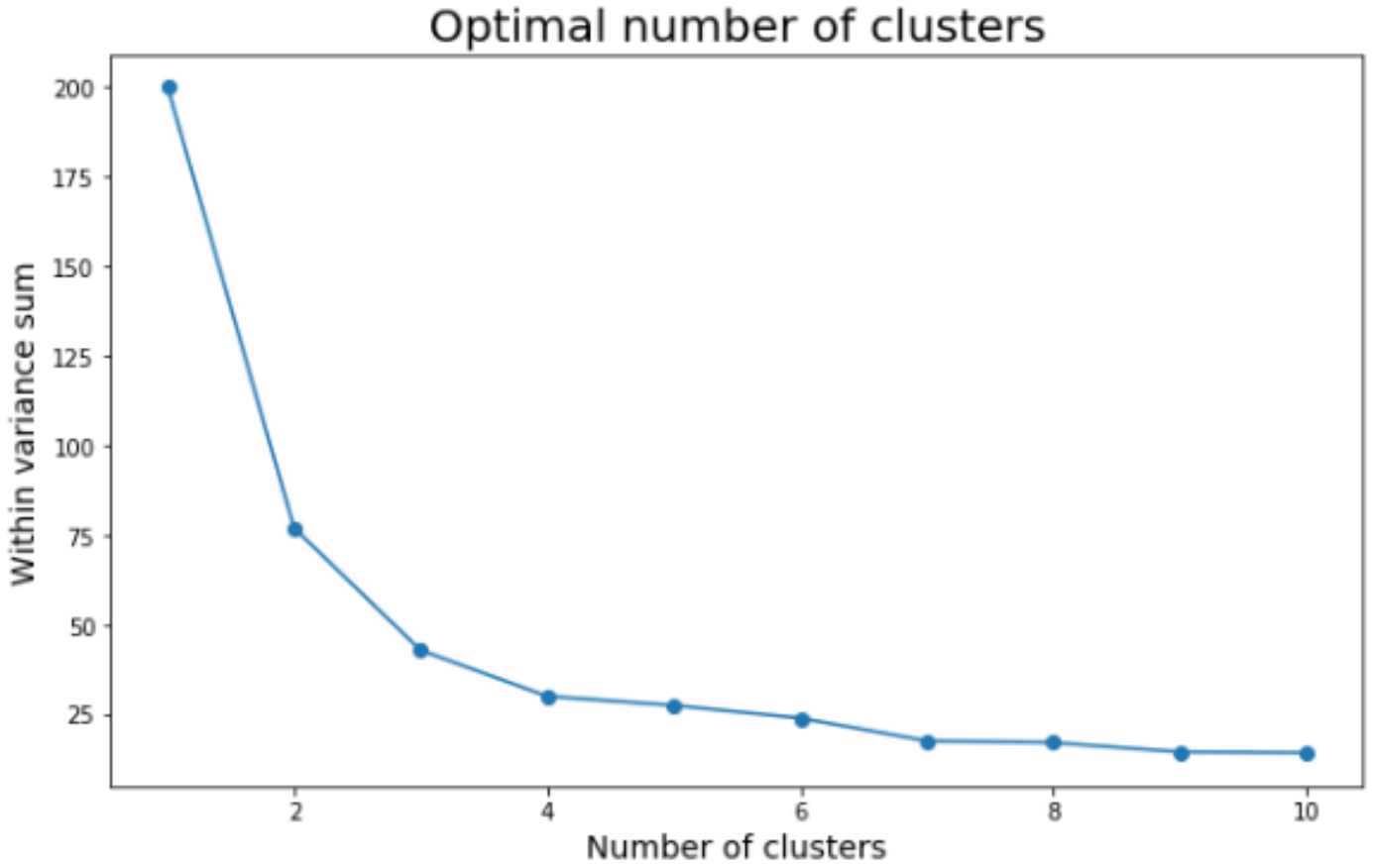
최적의 군집 개수 찾아보기.

- 거리제곱합을 통한 그림을 그려보자.

```
elbow = []
for nclusters in range(1, 11):
    i_kmeans_model = KMeans(n_clusters = nclusters,
                            init = "random", n_init = 1,
                            random_state = 10).fit(Iris_st)
    #inertia가 군집 내의 분산을 의미
    elbow.append(i_kmeans_model.inertia_)

# plot
plt.figure(figsize=(10,6))
plt.plot(range(1, 11), elbow, marker='o')
plt.title('Optimal number of clusters', fontsize=20)
plt.xlabel('Number of clusters', fontsize=14)
plt.ylabel('Within variance sum', fontsize=14)
plt.show()
```

6) 비계층적 군집분석-K평균 군집법

- 최적의 군집 개수 찾아보기.
  - 군집의 수가 3이 제일 좋아보임.



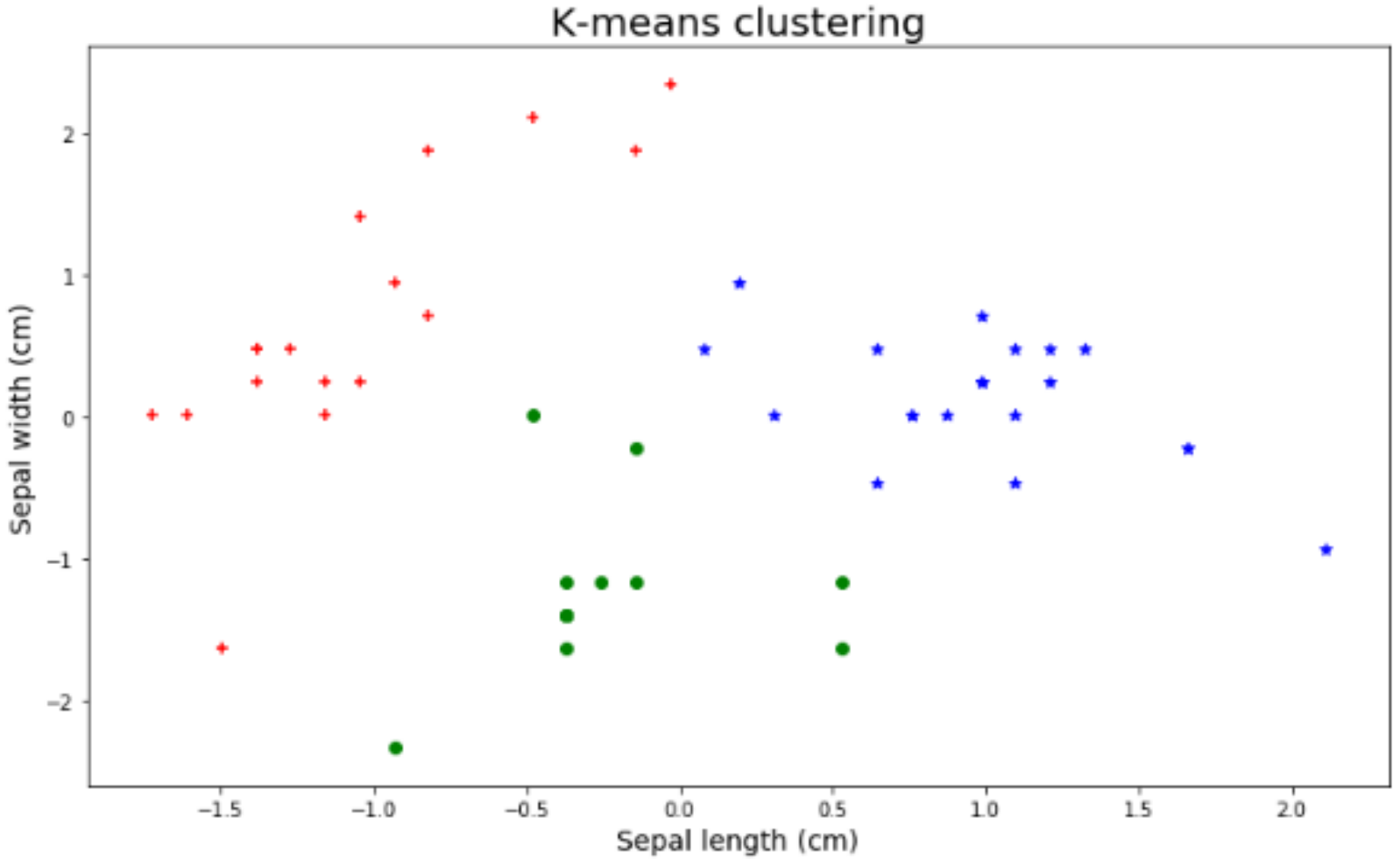
6) 비계층적 군집분석-K평균 군집법

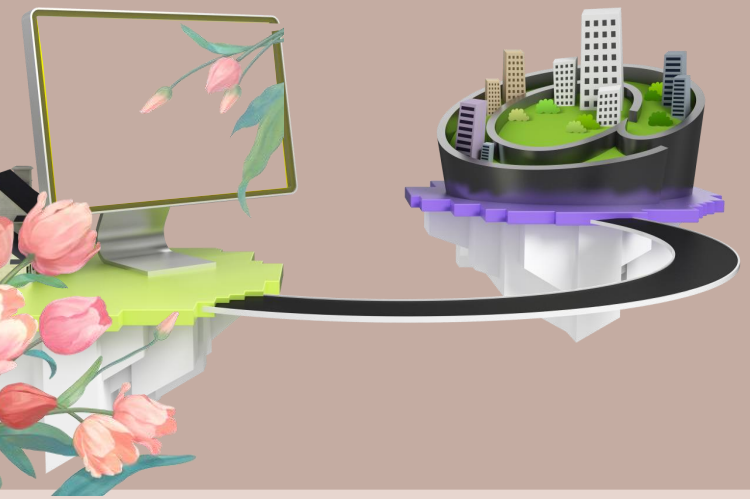
◆ 군집의 수가 3일 때 자료를 분석해보자.

```
# 군집 수 = 3
kmeans_model = KMeans(n_clusters = 3, init = "random",
                      n_init = 1, random_state = 10).fit(Iris_st)

cluster_results = kmeans_model.labels_
cluster_color = np.where(cluster_results == 1, 'red',
                         np.where(cluster_results == 2, 'green', 'blue'))
cluster_mark = np.where(cluster_results == 1, '+',
                        np.where(cluster_results == 2, 'o', '*'))

plt.figure(1, figsize=(12, 7))
for RowIdx in range(len(Iris_st)):
    plt.scatter(Iris_st['sepal_length'].iloc[RowIdx],
                Iris_st['sepal_width'].iloc[RowIdx],
                c=cluster_color[RowIdx],
                marker=cluster_mark[RowIdx])

plt.title('K-means clustering', fontsize=20)
plt.xlabel('Sepal length (cm)', fontsize=14)
plt.ylabel('Sepal width (cm)', fontsize=14)
plt.show()
```

6) 비계층적 군집분석-K평균 군집법

- 군집의 수가 3일 때 자료를 분석해보자.





다음시간안내

제09강

Decision tree