
파이썬 기반
빅데이터 처리
및 분석기술
Crayon

빅데이터 군집 분석

2 K-평균 군집 분석

- K-평균 소개
- 실습 : 숫자 데이터 분석

1. K-평균 알고리즘

1) 개요

◆ 최적의 군집화

- 군집 중앙은 해당 군집에 속하는 모든 점의 산술 평균이다.
- 각 점은 다른 군집의 중앙보다 자신이 속한 군집의 중앙에 더 가깝다.

1. K-평균 알고리즘

1) 개요

◆ 표준 패키지 불러오기

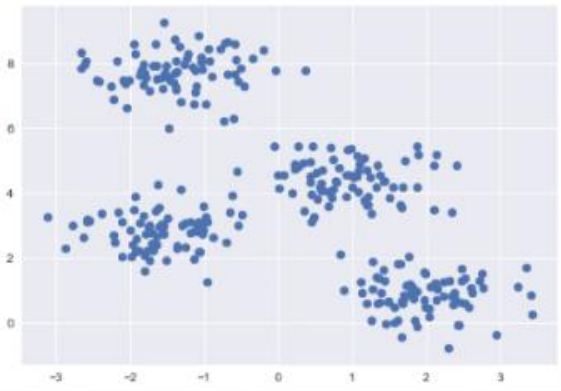
```
%matplotlib inline
import matplotlib.pyplot as plt
import seaborn as sns; sns.set()
import numpy as np
```

1. K-평균 알고리즘

1) 개요

◆ 데이터 생성

```
from sklearn.datasets.samples_generator import make_blobs
X, y_true =
make_blobs(n_samples=300, centers=4,cluster_std=0.60,random_state=0)
plt.scatter(X[:, 0], X[:, 1], s=50);
```

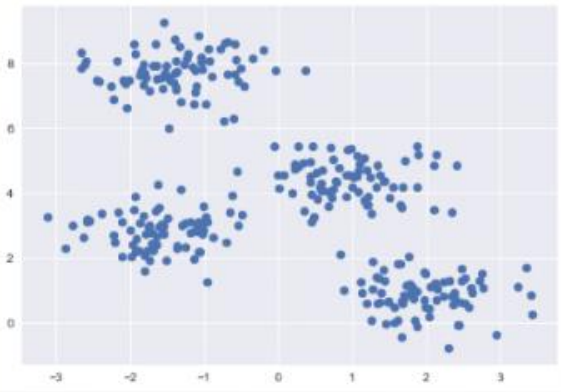


1. K-평균 알고리즘

1) 개요

◆ 데이터 생성

```
from sklearn.datasets.samples_generator import make_blobs
X, y_true =
make_blobs(n_samples=300, centers=4,cluster_std=0.60,random_state=0)
plt.scatter(X[:, 0], X[:, 1], s=50);
```



1. K-평균 알고리즘

2) 절차

◆ 모델 클래스 불러오기

- from sklearn.cluster import KMeans

1. K-평균 알고리즘

2) 절차

◆ 모델 클래스 불러오기

- from sklearn.cluster import KMeans

1. K-평균 알고리즘

2) 절차

◆ 모델 인스턴스 생성 및 초모수 설정

- kmeans = KMeans(n_clusters=4)

1. K-평균 알고리즘

2) 절차

◆ 모델 인스턴스 생성 및 초모수 설정

- kmeans = KMeans(n_clusters=4) → 중심점의 개수를 설정하는 초모수

K-평균 군집 분석은 군집 중심의 수를 변경하면서
최적의 군집을 만드는 군집 중심의 수를 찾는다.

1. K-평균 알고리즘

2) 절차

◆ 모델 적합

- kmeans.fit(X)

◆ 모델 적용

- y_kmeans = kmeans.predict(X)

1. K-평균 알고리즘

2) 절차

◆ 군집 중심 시각화

- centers = kmeans.cluster_centers_
  plt.scatter(centers[:, 0], centers[:, 1], c='black', s=200, alpha=0.5);

1. K-평균 알고리즘

2) 절차

◆ 군집 중심 시각화

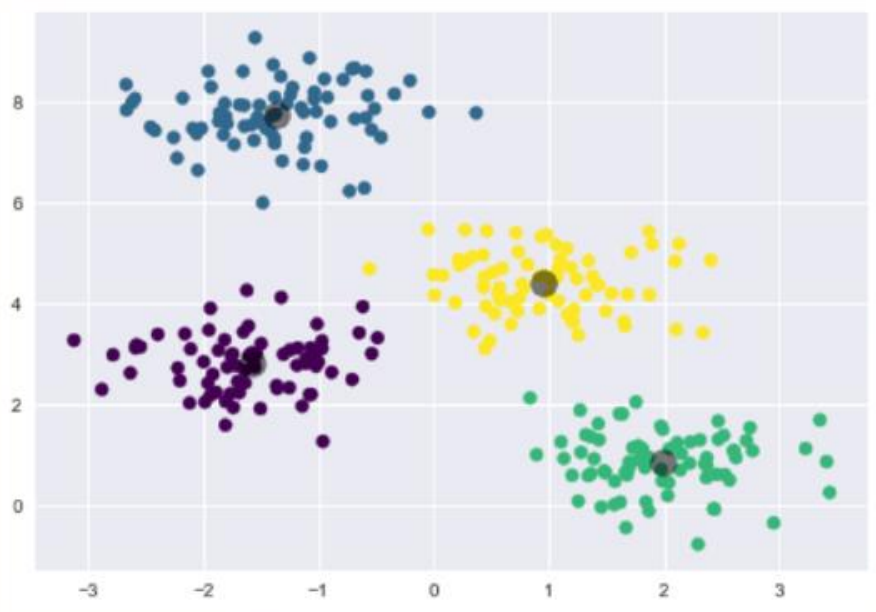
- centers = kmeans.cluster_centers_
  plt.scatter(centers[:, 0], centers[:, 1], c='black', s=200, alpha=0.5);

1. K-평균 알고리즘

2) 절차

◆ 데이터 시각화

- plt.scatter(X[:, 0], X[:, 1], c=y_kmeans, s=50, cmap='viridis');



1. K-평균 알고리즘

3) 원리

◆ 기댓값-최대화(E-M)

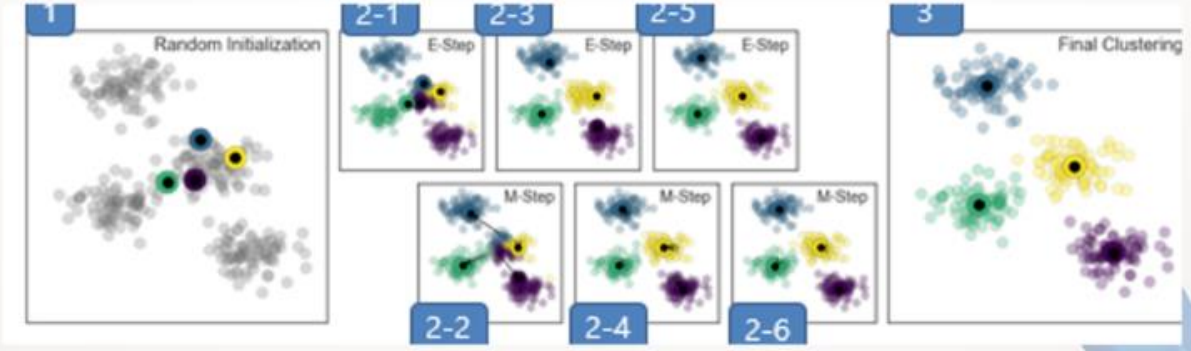
- 관측되지 않는 잠재 변수에 의존하는 확률 모델에서
  최대가능도를 갖는 모수의 추정값을 찾는 반복적인 알고리즘

1. K-평균 알고리즘

3) 원리

◆ 기댓값-최대화(E-M)

1	군집 중심을 임의로 추측한다.
2	군집 중심의 위치가 수렴할 때까지 다음을 반복한다. - 기댓값 단계 : 각 점을 가장 가까운 군집 중심에 할당한다. - 최대화 단계 : 할당된 데이터들의 산술평균을 통해 새로운 군집 중심을 찾는다.
3	수렴된 군집의 중심을 기준으로 군집을 할당한다.

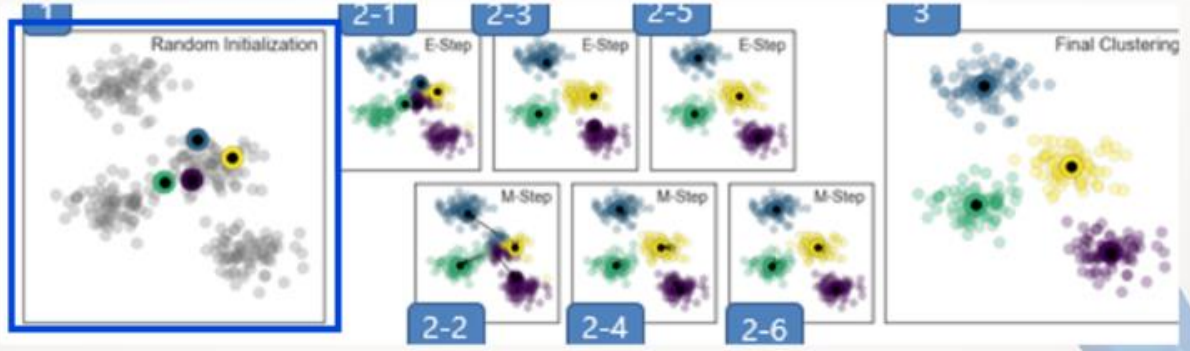


1. K-평균 알고리즘

3) 원리

◆ 기댓값-최대화(E-M)

1	군집 중심을 임의로 추측한다.
2	군집 중심의 위치가 수렴할 때까지 다음을 반복한다. - 기댓값 단계 : 각 점을 가장 가까운 군집 중심에 할당한다. - 최대화 단계 : 할당된 데이터들의 산술평균을 통해 새로운 군집 중심을 찾는다.
3	수렴된 군집의 중심을 기준으로 군집을 할당한다.

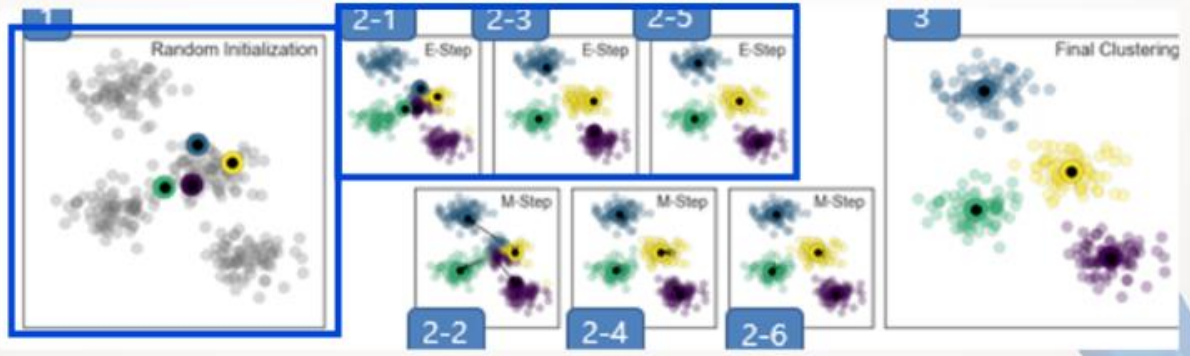


1. K-평균 알고리즘

3) 원리

◆ 기댓값-최대화(E-M)

1	군집 중심을 임의로 추측한다.
2	군집 중심의 위치가 수렴할 때까지 다음을 반복한다. - 기댓값 단계 : 각 점을 가장 가까운 군집 중심에 할당한다. - 최대화 단계 : 할당된 데이터들의 산술평균을 통해 새로운 군집 중심을 찾는다.
3	수렴된 군집의 중심을 기준으로 군집을 할당한다.

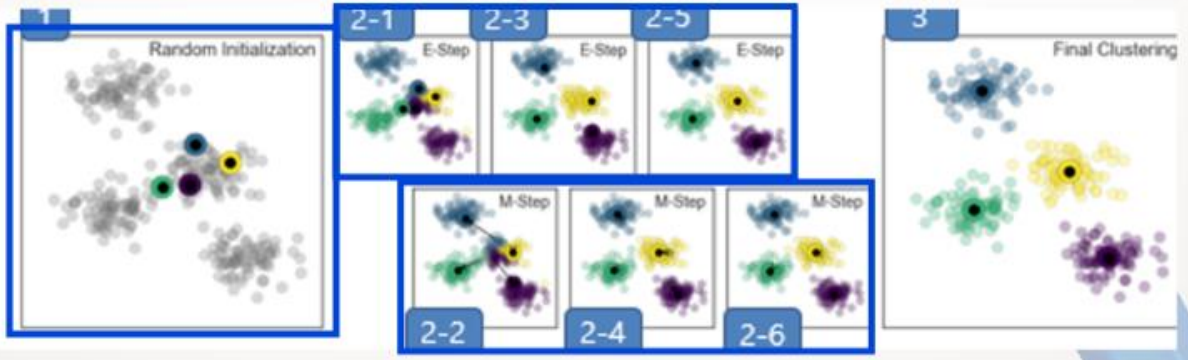


1. K-평균 알고리즘

3) 원리

◆ 기댓값-최대화(E-M)

1	군집 중심을 임의로 추측한다.
2	군집 중심의 위치가 수렴할 때까지 다음을 반복한다. - 기댓값 단계 : 각 점을 가장 가까운 군집 중심에 할당한다. - 최대화 단계 : 할당된 데이터들의 산술평균을 통해 새로운 군집 중심을 찾는다.
3	수렴된 군집의 중심을 기준으로 군집을 할당한다.

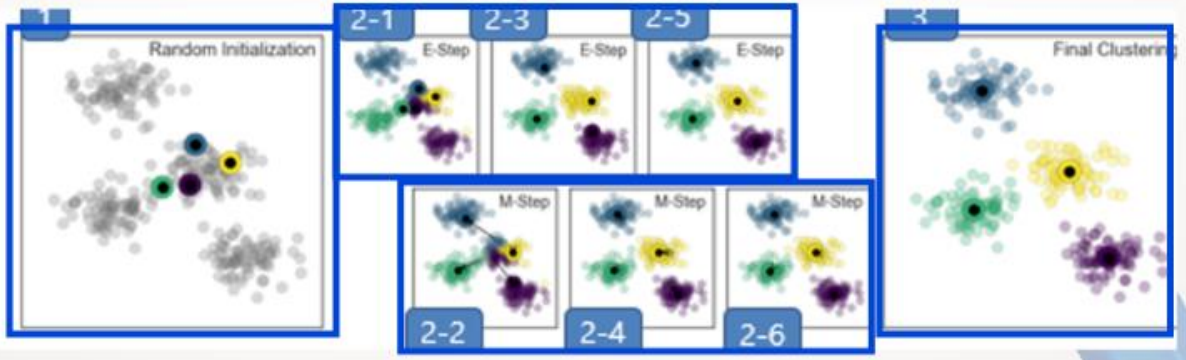


1. K-평균 알고리즘

3) 원리

◆ 기댓값-최대화(E-M)

1	군집 중심을 임의로 추측한다.
2	군집 중심의 위치가 수렴할 때까지 다음을 반복한다. - 기댓값 단계 : 각 점을 가장 가까운 군집 중심에 할당한다. - 최대화 단계 : 할당된 데이터들의 산술평균을 통해 새로운 군집 중심을 찾는다.
3	수렴된 군집의 중심을 기준으로 군집을 할당한다.



2. 실습

1) 숫자 데이터 분석

◆ 숫자 데이터 불러오기

- from sklearn.datasets import load_digits
  digits = load_digits()

2. 실습

1) 숫자 데이터 분석

◆ 숫자 데이터 시각화

```
for i in range(3):
  for j in range(5):
    ax[i][j].axis('off')
    ax[i][j].imshow(digits.data[i*4 + j].reshape(8, 8), cmap='binary', );
```

2. 실습

1) 숫자 데이터 분석

◆ 숫자 데이터 시각화

```
for i in range(3):
    for j in range(5):
        ax[i][j].axis('off')
        ax[i][j].imshow(digits.data[i*4 + j].reshape(8, 8), cmap='binary', );
```

2. 실습

1) 숫자 데이터 분석

◆ K-평균 군집 분석

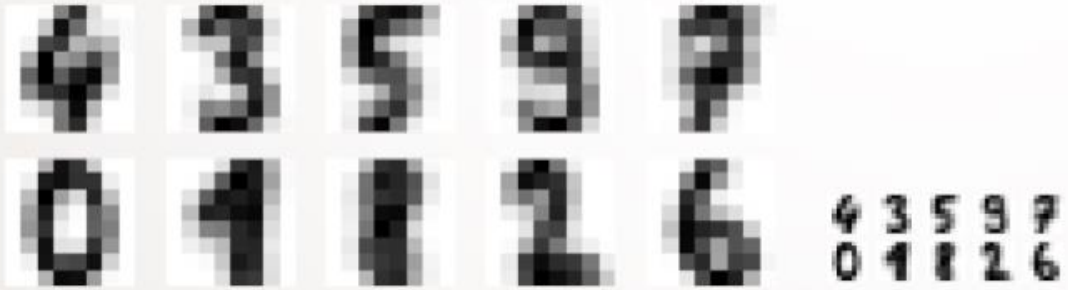
- ```
  from sklearn.cluster import KMeans
  kmeans = KMeans(n_clusters=10, random_state=0)
  clusters = kmeans.fit_predict(digits.data)
  kmeans.cluster_centers_.shape
  ```
  ```

2. 실습

1) 숫자 데이터 분석

◆ 군집 분석 시각화

```
fig, ax = plt.subplots(2, 5, figsize=(8, 3))
centers = kmeans.cluster_centers_.reshape(10, 8, 8)
for axi, center in zip(ax.flat, centers):
  axi.axis('off')
  axi.imshow(center, cmap=plt.cm.binary)
```



이번 시간에는

2 K-평균 군집 분석

- K-평균 소개
- 실습 : 숫자 데이터 분석

이번 시간에는

실습 참고 자료

- Colab 노트북 파일
- Scikit-Learn 공식 사이트 자료
  - https://scikit-learn.org/stable/user_guide.html

다음 시간에는

3 심화: 가우스 혼합 모델

- 가우스 혼합 모델 등장 배경
- 가우스 혼합 모델 실습