비지도 학습

데이터 전처리와 스케일 조정

- 여러가지 전처리 방법
  A. StandardScaler : 각 특성의 평균을 0, 분산을 1로 변경하여 모든 특성이 같은 크기를 가지게
  B. RobustScaler : 특성들이 같은 스케일을 갖게 된다는 통계적 측면에서 위와 비슷, 하지만 평균과 분산 대신 중간 값과 사분위 값을 사용
  C. MinMaxScaler : 모든 특성이 정확하게 0과 1사이에 위치하도록 데이터를 변경
  D. Normalizer : 특성 벡터의 유클리디안 길이가 1이 되도록 데이터 포인트를 조정
- 데이터 변환 적용
  SVM을 적용하려면 테스트 세트도 변환해야 한다.
  모든 스케일 모델은 항상 훈련 세트와 테스트 세트에 같은 변환을 적용해야 한다
  (무조건 훈련 세트부터 변환을 적용)
  scaler.transform(X_train), scaler.transform(X_test)

차원 축소, 특성 추출, 매니폴드 학습

- 주성분 분석(PCA)
  ➔ 특성들이 통계적으로 상관관계가 없도록 데이터셋을 회전시키는 기술
  회전한 뒤에 데이터를 설명하는 데 얼마나 중요하느냐에 따라 종종 새로운 특성 중 일부만 선택

  PCA를 사용하기 전에는 StandardScaler를 사용해 각 특성의 분산이 1이 되도록 데이터의 스케일을 조정 (올바른 주성분 방향을 찾기 위해)

  - 고유얼굴 특성 추출
    PCA는 특성 추출에 유용
    PCA를 사용 X : 얼굴의 유사도 측정을 위한 원본 픽셀 공간에서 거리 계산은 매우 나쁜 방법
    ⇨ PCA의 화이트닝 옵션을 사용해서 주성분의 스케일이 같아지도록 조정

    고유얼굴 주성분 : ex) 얼굴의 배치, 조명 등등

  PCA의 본질은 차원 축소

  ➔ 데이터를 회전시키고 분산이 작은 주성분을 덜어내는 것 (평균에 모여져 있는 것)
  ➔ 주성분은 가장 큰 분산의 방향을 차례대로 찾기 때문에 맨 처음 찾은 주성분이 재구성에 기여하는 정도가 가장 크고 나중으로 갈 수록 작음

- 비음수 행렬 분해 (NMF)

  PCA와의 차이

  ➔ PCA에서는 데이터의 분산이 가장 크고 수직인 성분을 찾았다면 NMF에서는 음수가 아닌 성분과 계수를 찾는다

  ➔ PCA는 재구성 측면에서 최선의 방향을 찾아주지만, NMF는 데이터에 있는 유용한 패턴을 찾는데 활용

  ➔ 성분들이 모두 양수값이여서 PCA보다 훨씬 더 얼굴 원형처럼 보임 (고유얼굴 분석)

- t-SNE를 이용한 매니폴드 학습

  높은 차원의 복잡한 데이터를 2차원에 차원 축소하는 방법이다. 높은 차원 공간에서 비슷한 데이터 구조는 낮은 차원에서 가깝게 대응하며, 비슷하지 않은 데이터 구조는 멀리 떨어져 대응된다.

  TSNE에는 transform 메소드가 없으므로 대신 fit_transform을 사용

  Ex) tsne.fit_transform(digits.data

군집

- k-평균 군집

  데이터의 어떤 영역을 대표하는 클러스터 중심을 찾는다. 그런 다음 클러스터에 할당된 데이터 포인터 평균으로 클러스터 중심을 다시 지정한다. 클러스터에 할당되는 데이터 포인트에 변화가 없을 때 알고리즘이 종료된다

  'n_clusters=?' 로 클러스터 수 조절

  - k-평균 알고리즘이 실패하는 경우

    k-평균은 모든 클러스터의 반경이 똑같다고 가정

    클러스터 중심 사이의 정확히 중간에 경계를 그린다

    ⇨ 원형이 아닌 클러스터를 구분하지 못한다

  - 벡터 양자화 또는 분해 메서드로서의 k-평균

    K-평균을 각 포인터가 하나의 성분으로 분해되는 관점으로 보는 것을 벡터 양자화

    데이터가 2차원이면 PCA나 NMF로 할 수 있는 것이 많지 않다

    복잡한 형태의 데이터셋을 다루기 위해서는 많은 클러스터를 사용하는 k-평균을 사용할 수 있다

  - K-평균의 단점

    1. 무작위 초기화를 사용하여 알고리즘의 출력이 난수 초기값에 따라 달라짐
    2. 클러스터의 모양을 가정하고 있어서 활용 범위가 비교적 제한적
    3. 찾으려 하는 클러스터의 개수를 지정해야 한다

sol) 엘보우 방법

⇨ 클러스터의 개수를 늘려가면서 k-평균의 이너셔가 감소가 완만해지는 지점을 찾아준다

- 병합 군집

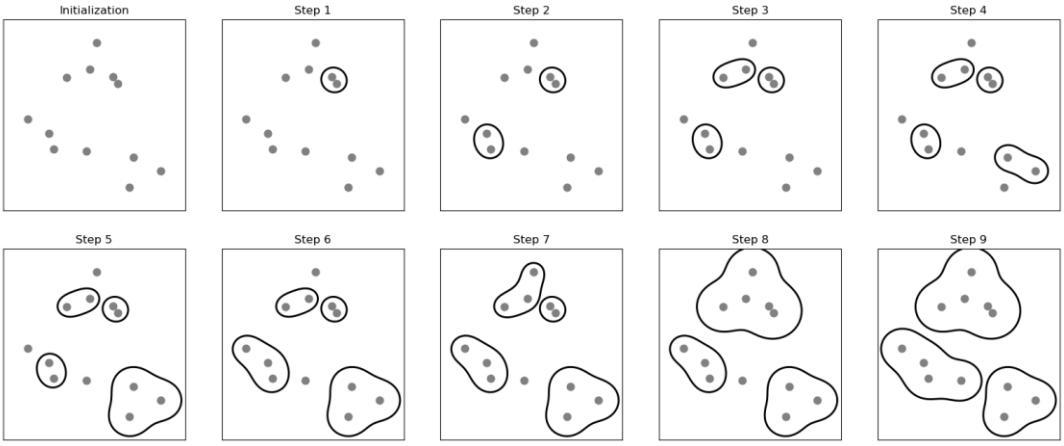
시작할 때 각 포인트를 하나의 클러스터로 지정하고, 그 다음 어떤 종료조건을 만족할 때까지 가장 비슷한 두 클러스터를 합쳐나간다

ward : 모든 클러스터 내의 분산을 가장 작은 증가시키는 두 클러스터를 합친다

average : 클러스터 포인트 사이의 평균거리가 가장 짧은 두 클러스터를 합친다

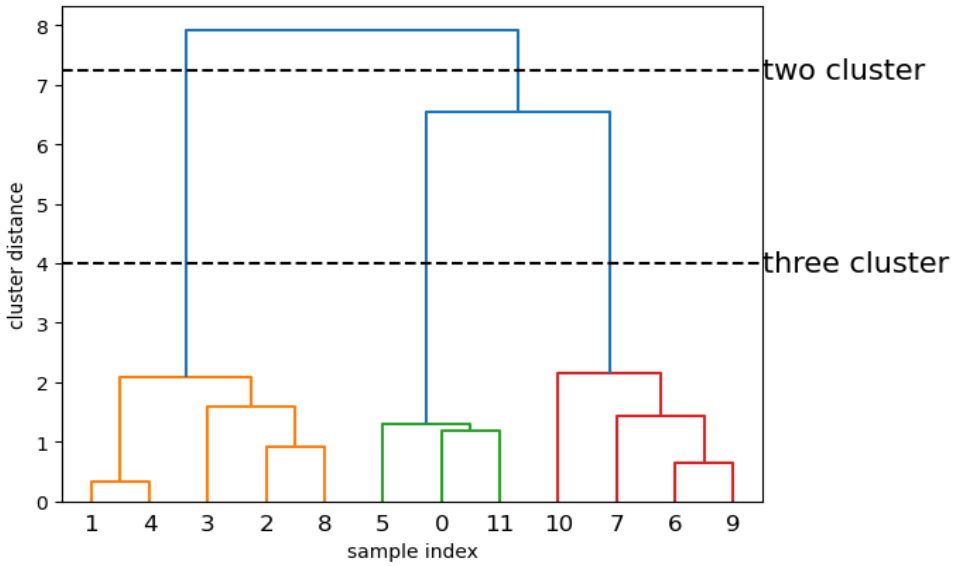
complete : 클러스터 포인트 사이의 최대 거리가 가장 짧은 두 클러스터를 합친다

AgglomerativeClustering(n_clusters=3)



■ 계층적 군집과 덴드로그램

Ex)



[1,4] : 샘플개수(12) 보다 작음 => 1과 4를 자식 노드로 가짐

[5,15] : 샘플개수(12) 보다 큼 => 15 - 12 = 노드 인덱스 3을 자식으로 가짐([0, 11])

- DBSCAN

  클러스터의 개수를 미리 지정할 필요가 없다

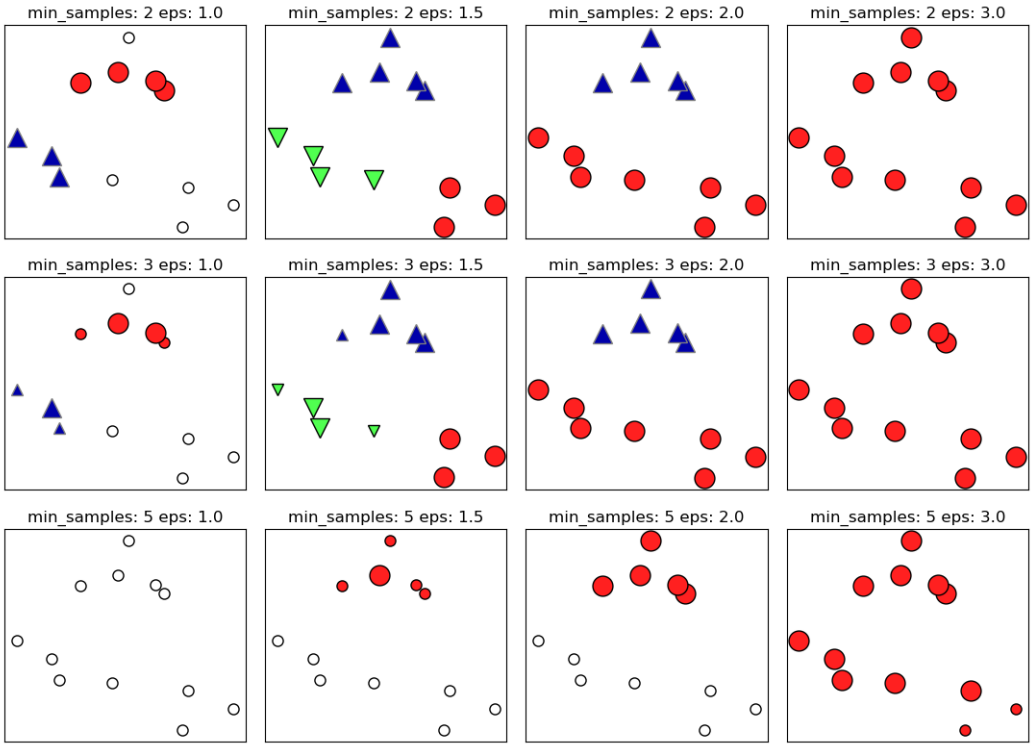
  병합 군집이나 k-평균보다는 다소 느리지만 비교적 큰 데이터셋에도 적용할 수 있음

  밀집지역을 찾는다

  한 데이터 포인트에서 eps 거리 안에 데이터가 min_samples 개수만큼 들어있으면 이 데이터 포인트를 핵심 샘플로 분류

  # min_samples보다 적으면 '잡음 레이블'로 분류

  Ex)



  Esp를 증가시키면 하나의 클러스터에 더 많은 포인트가 포함, 여러 클러스터를 하나로 합치게 만듬

  min_samples로 키우면 핵심 포인트 수가 줄어들며 잡음 포인트가 늘어난다

  적절한 esp 값을 쉽게 찾기

  ➔ Standardscaler나 minmaxscaler로 모든 특성의 스케일을 비슷하게 만든다

- 군집 알고리즘의 비교와 평가
  - ■ 타깃 값으로 군집 평가하기

    군집 알고리즘의 결과를 실제 정답 클러스터와 비교하여 평가할 수 있는 지표들이 있다

    ➔ ARI : 1(최적일 때) 와 0(무작위로 분류될 때)

    adjusted_rand_score()
  - ■ 타깃 값 없이 군집 평가하기

    군집 알고리즘을 적용할 때 보통 그 결과와 비교할 타깃 값이 없음

실루엣 계수 : 타깃값이 필요 없는 군집용 지표

silhouetee_score()

■ 얼굴 데이터 셋으로 군집 알고리즘 비교

➢ DBSCAN

잡음 포인트 : 특이한 것

⇨ 이상치 검출

Eps 값에 따라 클러스터 수, 크기 확인

➢ K-mean

k-평균은 DBSCAN처럼 잡음 포인트 개념이 없이 모든 포인트를 구분하기 때문에 중심에서 먼 포인트들은 클러스터 중심과 많이 달라 보인다

➢ 병합 군집으로 얼굴 데이터셋 분석

Ex) 40개의 클러스터 중 39번 클러스터에 속한 얼굴사진