3-3 군집 분석

군집분석(cluster analysis)

- 각 개체에 대해 관측된 여러 개의 변수 (x1,x2,⋯ ,xp) 값들로부터
  - n개의 개체를 유사한 성격을 가지는 몇 개의 군집으로 집단화하고,
  - 형성된 군집들의 특성을 파악하여 군집들 사이의 관계를 분석하는 다변량분석 기법
- (대표적인 **비지도 학습**) 군집 분석에 이용되는 다변량 자료는 별도의 반응변수가 요구되지 않으며,
  - 오로지 개체들 간의 유사성(similarity)에만 기초하여 군집을 형성
- 군집 분석은 이상값 탐지에도 사용되며, 심리학, 사회학, 경영학, 생물학 등 다양한 분야에 이용

군집화의 방법

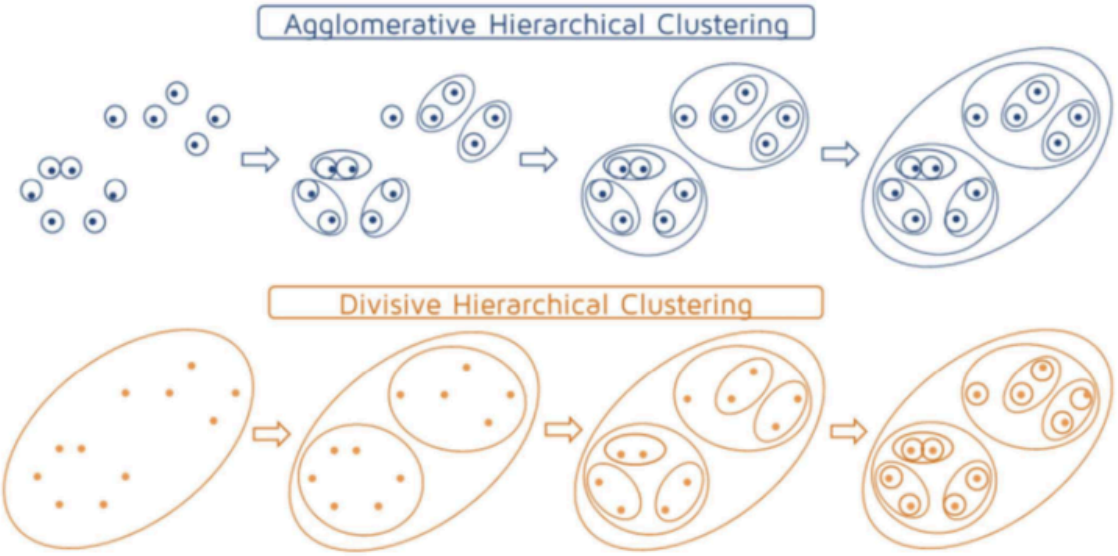
- 계층적 군집
- 분할(또는 분리(partitioning)) 군집
- 밀도-기반 군집
- 모형-기반 군집
- 격자-기반 군집
- 커널-기반 군집
- SOM(Self-Organizing Maps) 방법 등

계층적 군집(hierarchical clustering)

- 가장 유사한 개체를 묶어 나가는 과정을 반복하여 원하는 개수의 군집을 형성하는 방법
  - 보통 계통도 또는 덴드로그램(dendrogram)의 형태로 결과가 주어지며
  - 각 개체는 하나의 군집에만 속하게 된다.
- 개체간의 유사성(또는 거리)에 대한 다양한 정의가 가능
  - 군집간의 연결법(최단연결법, 최장연결법, 평균연결법, 중심연결법, 와드연결법)에 따라 군집의 결과가 달라질 수 있음

계층적 군집을 형성하는 방법

- 병합적(agglomerative) 방법: bottom-up으로 작은 군집으로부터 출발하여 군집을 병합해 나가는 과정
    - 한 개의 항목으로 시작하여 군집을 형성해 나가는 매 단계 마다 모든 그룹 쌍 간의 거리를 계산하여 가까운 순으로 병합을 수행
    - 이 과정을 한 개 그룹만 남을 때까지 혹은 종료의 조건이 될 때 까지 반복
    - 여기에서 그룹 혹은 항목간의 상대적 거리가 가까울수록 유사성(similarity)이 높다고 말할 수 있음
- 분할적(divisive) 방법: top-down 방식으로 큰 군집으로부터 출발하여 군집을 분리해 나가는 방법



출처 https://blog.naver.com/luexr/223463130733?trackingCode=rss

📏 계층적 군집의 거리 측정 방법

요약하자면,

계층적 군집에서 거리(또는 유사도)를 측정하는 방법에는 대표적으로 다섯 가지가 있다.

이는 두 개의 군집 간의

"가장 가까운 점", **"가장 먼 점"**, **"평균 거리"**, **"중심 거리"**, **"군집 내 분산 증가량"** 등을 기준으로 군집 간 거리를 정의하는 방식들이야.

이러한 기준에 따라 같은 데이터여도 전혀 다른 군집 결과가 나올 수 있지.

거리 측정 방법

1. 최단 연결법 (Single Linkage)

- 두 군집 간 가장 가까운 두 점 사이의 거리를 군집 간 거리로 정함.
- **장점**: 연쇄적 연결로 길쭉하고 불규칙한 모양의 군집 탐지에 유리.
- **단점**: 잡음이나 이상치에 민감하고, 군집이 연쇄적으로 연결되는 chaining 현상이 발생할 수 있음.

2. 최장 연결법 (Complete Linkage)

- 두 군집 간 가장 먼 두 점 사이의 거리를 군집 간 거리로 정함.
- **장점**: 군집 내 응집력이 강한 조밀한 군집 생성.
- **단점**: 군집의 크기 차이나 밀도 차이에 민감.

3. 평균 연결법 (Average Linkage, UPGMA)

- 두 군집에 속한 모든 점 쌍의 거리 평균을 군집 간 거리로 정함.
- **장점**: 극단값의 영향을 덜 받고, 비교적 안정적인 군집 생성.
- **단점**: 계산량이 많고, 해석이 직관적이지 않을 수 있음.

4. 중심 연결법 (Centroid Linkage)

- 군집의 중심(centroid) 간 거리로 군집 간 거리 측정.
- **장점**: 중심점을 통해 직관적인 해석 가능.
- **단점**: 두 군집의 병합 후 중심이 군집 외부에 위치할 수 있어 이상한 결과 유발 가능.

5. 와드 연결법 (Ward's Method)

- 두 군집을 합쳤을 때 **군집 내 분산(군집내의 오차 제곱합)의 증가량**을 거리로 정의.
- **장점**: 군집 내 분산을 최소화하여 크기가 비슷하고 조밀한 군집 생성.
- **단점**: 계산 복잡도 높음, 군집 간 거리 해석이 직관적이지 않음.

🧮 보너스: 수식으로 본 개요

방법	거리 정의 방식	특성
Single	$\min \|x - y\|$	가장 가까운 점들
Complete	$\max \|x - y\|$	가장 먼 점들
Average	$\frac{1}{mn} \sum_{i=1}^{m} \sum_{j=1}^{n} \|x_i - y_j\|$	모든 쌍 평균
Centroid	$\|c_1 - c_2\|$	중심점 간 거리
Ward	SSE 증가량	군집 내 제곱합 최소화

📏 다양한 거리 정의 방법 정리

군집 분석이나 유사도 기반 기법에서 "거리(distance)"는 개체 간의 유사성/차이를 수치로 표현한 핵심 개념이다.

데이터의 특성과 목적에 따라 적합한 거리 정의 방법을 선택해야 하며, 대표적으로 **유클리디안 거리, 맨해튼 거리, 민코프스키 거리, 코사인 거리, 자카드 거리, 해밍 거리** 등이 있다.

📏 다양한 거리 정의 방법 정리

1. 유클리디안 거리 (Euclidean Distance)

- 가장 대표적인 거리 개념, 두 점 사이의 직선 거리.
- 수식:

$$d(x, y) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}$$

- **특징**:
  - 연속형 수치 데이터에 적합
  - 변수 간 스케일에 민감 → 표준화 필요

2. 맨해튼 거리 (Manhattan Distance)

- 축을 따라 직각 방향으로만 이동한 거리 (L1 노름)
- 수식:

$$d(x, y) = \sum_{i=1}^{n} |x_i - y_i|$$

- **특징**:
  - 정사각형 블록 구조나 절대 거리 차이에 민감한 경우 사용
  - 이상값 영향 덜 받음

3. 민코프스키 거리 (Minkowski Distance)

- 유클리디안과 맨해튼 거리의 일반화 형태
- 수식:

$$d(x, y) = \left(\sum_{i=1}^{n} |x_i - y_i|^p \right)^{1/p}$$

- **특징**:
  - p=1p = 1p=1이면 맨해튼 거리, p=2p = 2p=2이면 유클리디안 거리
  - p를 조절하여 다양한 거리 측정 가능

4. 코사인 거리 (Cosine Distance)

- 벡터 간 방향의 차이(코사인 유사도)를 바탕으로 측정
- 수식 (거리):

$$d(x, y) = 1 - \frac{x \cdot y}{\|x\| \|y\|}$$

- **특징**:
  - 문서 유사도, 고차원 희소 벡터에서 효과적
  - 크기보다는 방향 중시

5. 자카드 거리 (Jaccard Distance)

- 두 집합의 공통 원소 비율에 기반한 거리
- 수식:d(A,B)=

$$d(A, B) = 1 - \frac{|A \cap B|}{|A \cup B|}$$

- **특징**:
  - 이진 데이터나 집합형 데이터에 적합
  - 예: 영화 취향, 태그 일치도 등

6. 해밍 거리 (Hamming Distance)

- 동일 길이 이진 문자열 간 서로 다른 위치의 개수
- 수식:

$$d(x, y) = \sum_{i=1}^{n} 1_{x_i \neq y_i}$$

- **특징**:
  - 이진 벡터 또는 문자열 비교에 유리
  - 예: 유전자 염기서열, 오류 검출 등

📊 거리 선택 시 고려 사항

고려 항목	추천 거리
수치형 데이터	유클리디안, 맨해튼
이진/범주형 데이터	자카드, 해밍
고차원 희소 벡터	코사인 거리
이상치에 민감할 경우	맨해튼, 민코프스키(p<2)

연속형

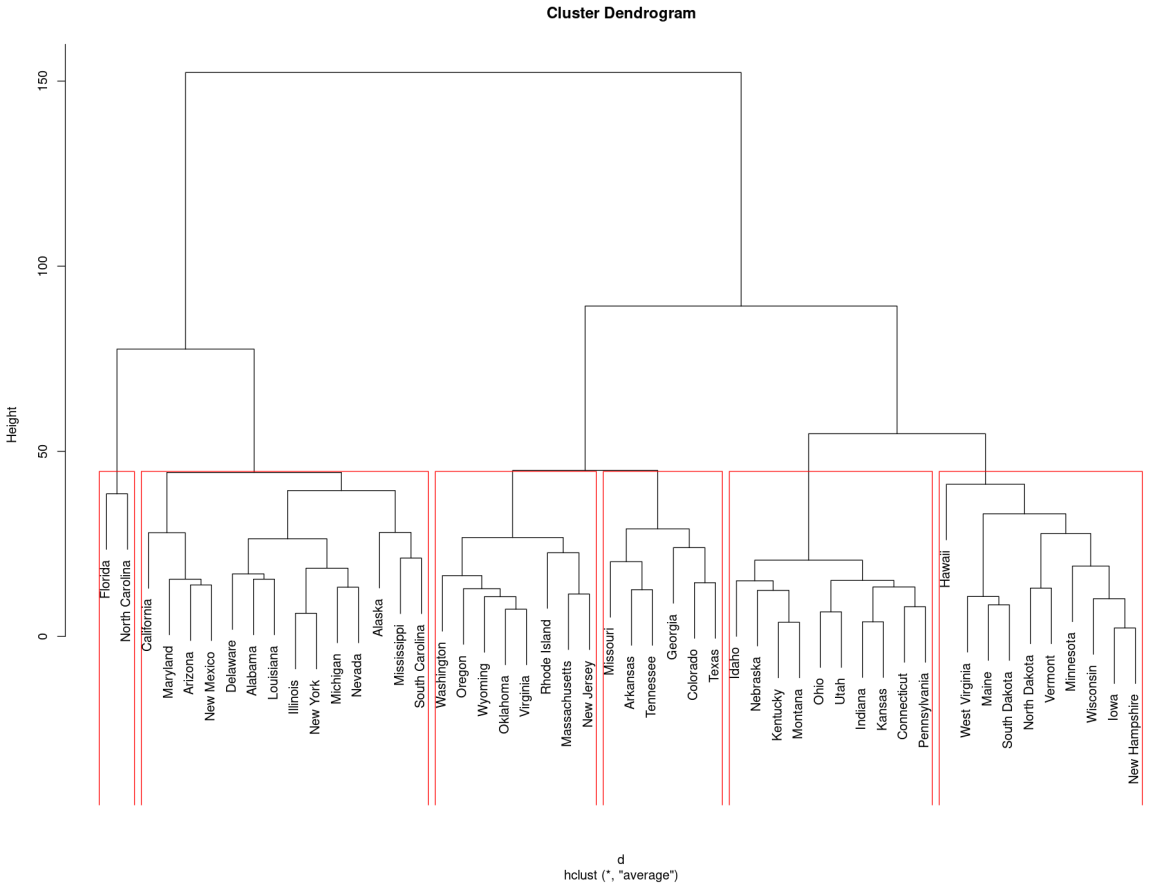
유클리디안 거리	두 점 사이의 거리	$d(x, y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}$
맨하튼 거리	변수들의 차이의 합	$d(x,y) = \sum_{i=1}^{n} \lvert x_i - y_i \rvert$
체비셰프 거리	변수 간 거리 차이 중 최댓값	$d(x, y) = max \lvert x_i - y_i \rvert$
표준화 거리	유클리디안 거리를 표준편차고 나눔	$d = \sqrt{\sum_{i=1}^{n}\left(\frac{x_i - \mu_i}{\sigma_i}\right)^2}$
마할라노비스 거리	표준화 거리에서 변수 간 상관성까지 고려	$d = \sqrt{(x-\mu)^T S^{-1}(x-\mu)}$
민코프스키 거리	유클리디안 거리와 맨하튼 거리를 한번에 표현	$d = \left(\sum_{i=1}^{n} \lvert x_i - y_i \rvert^p\right)^{\frac{1}{p}}$

범주형

단순 일치 계수	두 객체 i와 j 간의 상이성을 불일치 비율로 계산
자카드 지수	두 집합 사이의 유사도를 측정
코사인 유사도	크기가 아닌 방향성을 측정

출처 https://diversity-is-right.tistory.com/107

덴드로그램: 계층적 군집 분석 결과 시각화



k-평균 군집(k-means clustering)

원하는 군집 수만큼(k개) 초기값을 지정하고, 각 개체(데이터)를 가까운 초기값에 할당하여 군집을 형성한 뒤, 각 군집의 평균을 재계산하여 초기값을 갱신한다. 갱신된 값에 대해 위의 할당과정을 반복하여 k개의 최종군집을 형성한다.

k-평균군집의 절차(알고리즘)는 다음과 같다.

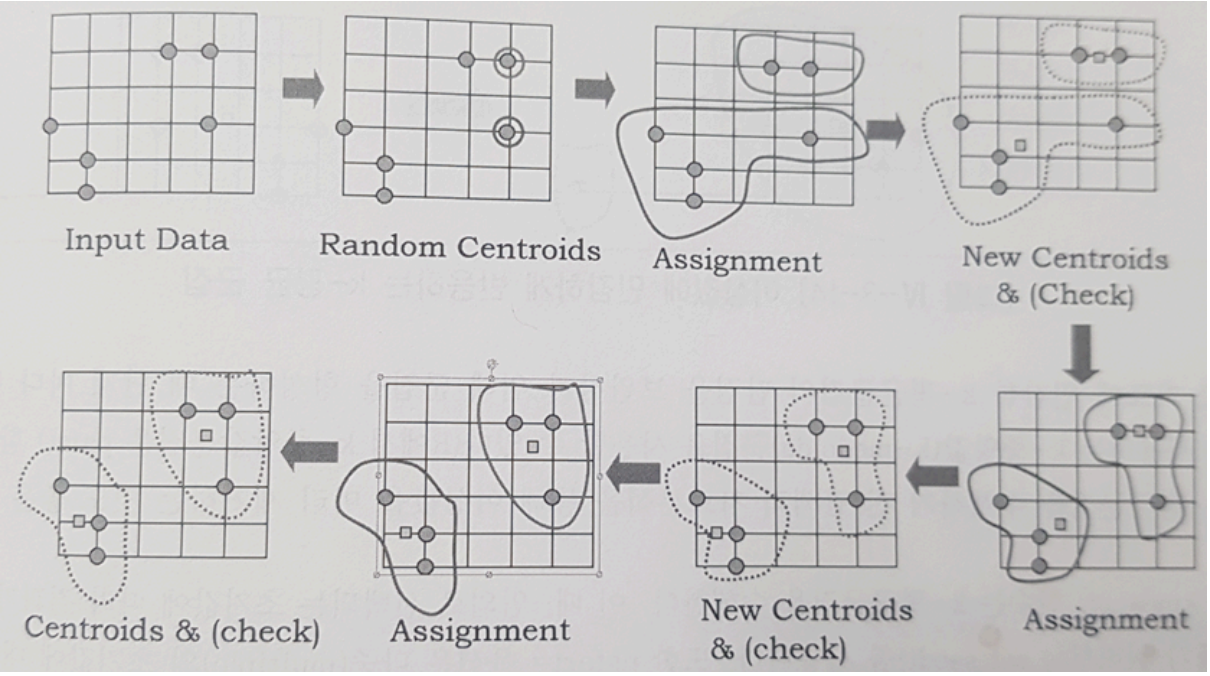
- 단계1. 초기 (군집의 중심)으로 k개의 객체를 임의로 선택한다.
- 단계2. 각 자료를 가장 가까운 군집 중심에 할당한다.
- 단계3. 각 군집 내의 자료들의 평균을 계산하여 군집의 중심을 갱신(update)한다.
- 단계4. 군집 중심의 변화가 거의 없을 때(또는 최대 반복수)까지 단계2와 단계3을 반복한다.

위의 단계2는 자료들의 군집의 중심점(평균)으로부터의 오차제곱합

$$E = \sum_{i=1}^{k} \sum_{x \in C_i} (x - \bar{x}_i)^2, \quad \bar{x}_i = \frac{1}{n} \sum_{x \in C_i} x$$

이 최소가 되도록 각 자료를 할당하는 과정이다.

아래의 그림은 k-평균군집의 절차



k-평균군집에서 군집의 수(k)는 미리 정해 주어야 하며, k개의 초기 중심값은 임의로 선택될 수 있으나,
자료값 중에서 무작위로 선택하는 것이 보다 편리할 것이다.

- 다만 초기 중심점들은 서로 멀리 떨어져 있는 것이 바람직하며, 초기값에 따라 군집 결과가 크게 달라질 수 있다.

k-평균군집은 군집의 매 단계마다 군집 중심으로부터의 오차제곱합을 최소화하는 방향으로 군집을 형성해 나가는(부분 최적화를 수행하는) 탐욕적(greedy) 알고리즘으로 간주될 수 있으며,

- 안정된 군집은 보장하나 전체적으로 최적이라는 것은 보장하지 못한다.

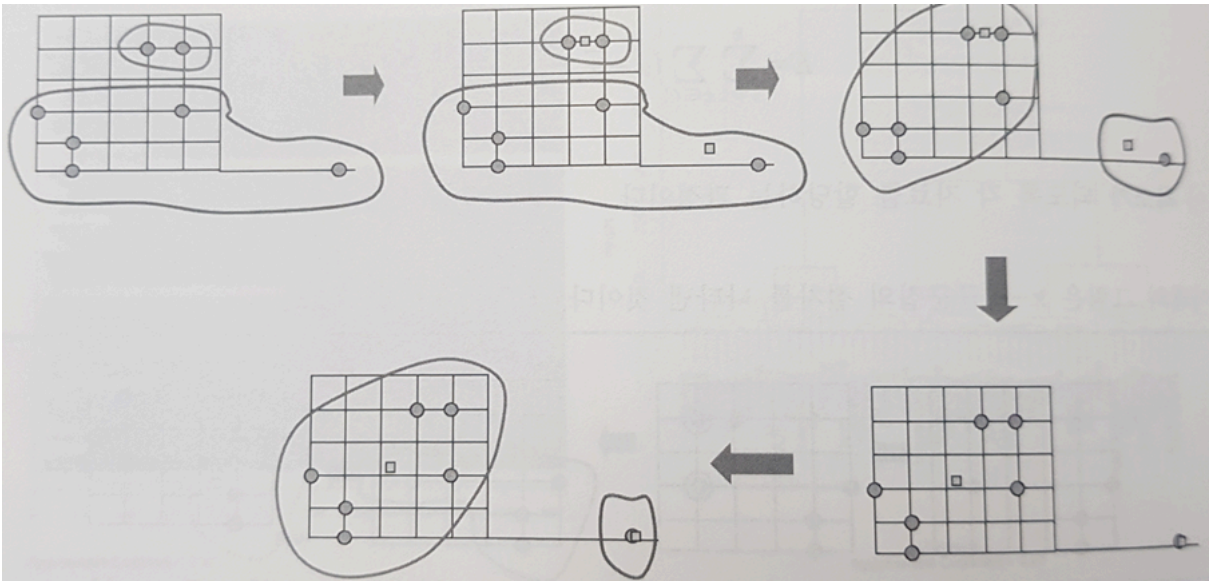
k-평균군집은 알고리즘이 단순하며, 빠르게 수행되며 계층적 군집보다 많은 양의 자료를 다룰 수 있으며,
평균 또는 거리 계산에 기반하므로 모든 변수가 연속적이어야 한다.

단점

- 잡음이나 이상값에 영향을 많이 받으며(군집의 중심을 계산하는 과정에서),
- 볼록한 형태가 아닌(non-convex) 군집(예를 들어, U-형태의 군집)이 존재할 경우에는 성능이 떨어진다.

이상값에 영향을 많이 받는 사례

- 아래의 그림은 k-평균군집이 이상값 자료에 대해 민감하게 반응하는 과정을 보여준다.



이상값 자료에 민감한 k-평균군집의 단점을 보완하기 위해 군집을 형성하는 매 단계마다 평균 대신
중앙값을 사용하는 k-중앙값(k-medoids)군집을 사용할 수 있다

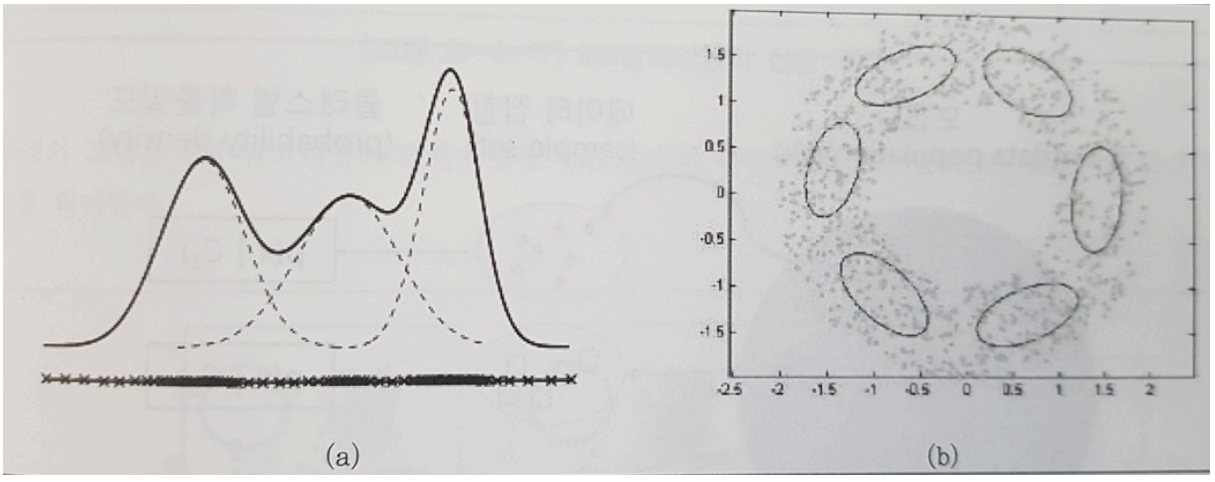
- R에서 k-중앙값군집은 pam() 함수를 이용

k-평균군집을 수행하기 전 탐색적 자료분석을 통해 이상값을 미리 제거하는 것도 좋은 방법

혼합 분포 군집(mixture distribution clustering)

모형-기반(model-based)의 군집 방법으로,

- 데이터가 k개의 모수적 모형(흔히 정규분포 또는 다변량 정규분포를 가정함)의 가중합으로 표현되는 모집단 모형으로부터 나왔다는 가정하에서 모수와 함께 가중치를 자료로부터 추정하는 방법을 사용
    - k개의 각 모형은 군집을 의미하며, 각 데이터는 추정된 k개의 모형 중 어느 모형으로부터 나왔을 확률이 높은지에 따라 군집의 분류가 이루어진다.
    - 흔히 혼합모형에서의 모수와 가중치의 추정(최대가능도추정)에는 EM 알고리즘이 사용된다.
    - 아래의 그림은 혼합분포모형을 통해 설명될 수 있는 데이터의 형태를 나타낸다.
    - 아래 그림의 (a)는 자료의 분포형태가 다봉형의 형태를 띠므로 단일 분포로의 적합은 적절하지 않으며, 대략 3개 정도의 정규분포의 결합을 통해 설명될 수 있을 것으로 생각할 수 있다.
    - (b)의 경우에도 여러 개의 이변량 정규분포의 결합을 통해 설명될 수 있을 것이다.
        - 두 경우 모두 반드시 정규분포로 제한할 필요는 없다.



EM(Expectation Maximizaion) 알고리즘의 정의와 특징

- 혼합모형에서 모수와 가중치의 추정을 위해 사용되는 알고리즘
- 미지의 분포 파라미터를 주어진 데이터를 가지고 예측하고

    - 그 예측값을 다시 주어진 데이터를 기반으로 기대치를 최대화시키는 파라미터를 구하는 과정을 반복
- 초기 클러스터의 개수를 정해줘야 함
- k-fold cross validation으로 적절한 클러스터 개수를 찾을 수 있음

혼합 분포 군집(Mixture Distribution Clustering) 요약

혼합 분포 군집은 데이터를 여러 개의 확률 분포(주로 정규분포)의 혼합으로 보고, 각 데이터가 어느 분포(=군집)에 속할 가능성이 높은지를 추정하는 **모형 기반 군집 방법**

- 주로 **EM 알고리즘**을 사용해서 분포의 모수와 군집 소속 확률을 추정

---

1. 개념과 기본 원리

- **혼합 분포 군집**은 데이터를 여러 개의 확률 분포(모형)가 섞여 있는 것으로 보고,
    - 각각의 분포가 하나의 **군집(cluster)**를 나타낸다고 가정
- 주로 **정규분포**나 다변량 정규분포(Gaussian)를 사용하여 데이터를 설명
- 각 데이터는 **k개의 분포 중 하나에서 생성**되었을 확률을 가짐
    - 이 확률로 **군집 소속**을 결정

2. 수학적 표현

- 데이터 x는 다음과 같은 혼합 분포에서 생성되었다고 가정함:

$$P(x) = \sum_{i=1}^{k} \pi_i \cdot P(x|\theta_i)$$

- πi: i번째 분포의 가중치(군집에 속할 확률)
- P(x ∣ θi): i번째 분포(예: 정규분포)의 확률 밀도 함수
- θi: 각 분포의 모수(평균, 공분산 등)

3. 학습 방법 - EM 알고리즘

- **E-step**: 각 데이터가 각 군집에 속할 사후 확률(posterior probability)을 계산
- **M-step**: 그 확률을 바탕으로 각 분포의 모수(parameter)와 **가중치**를 업데이트
- 이 과정을 반복해서 수렴할 때까지 군집과 분포를 추정함
- EM은 확률적이며, 군집 중심을 초기화에 따라 다르게 찾을 수 있음

4. 장점과 한계

- **장점**:
  - 원형(cluster shape)이 아닌 **타원형이나 다양한 형태**의 군집도 잘 분리 가능
  - 군집의 불확실성(soft assignment)을 제공함(k-means와 달리 이 데이터가 어느 군집에 속할 확률도 나옴)
- **한계**:
  - 모델 수 k를 **사전에 정해야** 함
  - 초기값에 따라 수렴 결과가 다를 수 있음 (지역 최적점 문제)
  - 계산 비용이 큼 (특히 고차원 데이터에서)

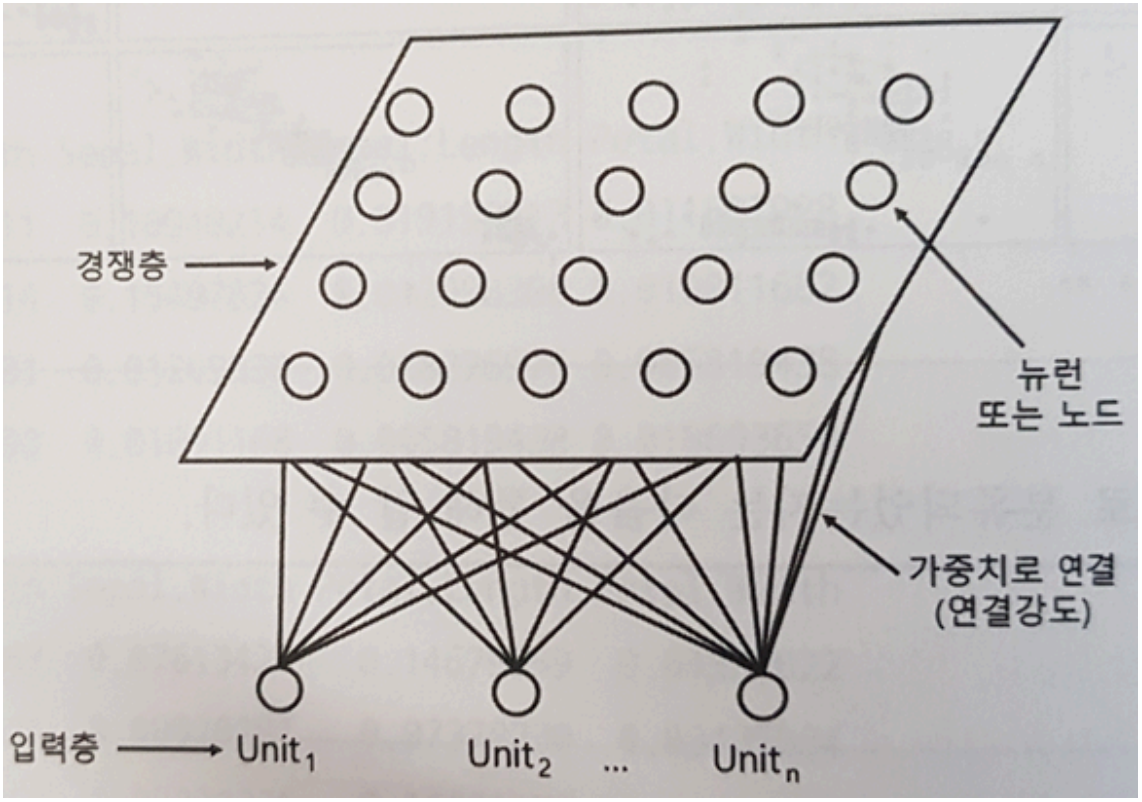
SOM(Self-Organizing Maps, 자기조직화지도)

코호넨(Kohonen, 1990; Kohonen, 1995; Kohonen et al., 1996)에 의해 개발되어 코호넨 맵(Kohonen Maps)이라고도 알려져 있음

- SOM은 비지도 신경망(unsupervised neural network)으로 고차원의 데이터를 이해하기 쉬운 저차원의 뉴런(neuron)으로 정렬하여 지도(map)의 형태로 정량화
- 이러한 정량화는 입력 변수의 위치 관계를 그대로 보존한다는 특징
  - 다시 말해 실제 공간의 입력 변수가 가까이 있으면, 지도상에서도 가까운 위치에 있게 됨
  - 이러한 SOM의 특징으로 인해 입력 변수의 정보와 그들의 관계가 지도상에 그대로 나타냄
- SOM 모델은 다음 그림과 같이 두 개의 인공신경망 층(입력층, 경쟁층)으로 구성
  - 하나는 입력층(input layer: 입력벡터를 받는 층)
    - 입력층은 입력 변수의 개수와 동일한게 뉴런 수가 존재

    - 다른 하나는 2차원 격자(grid)로 구성된 경쟁층(competitive layer: 입력 벡터의 특성에 따라 벡터가 한 점으로 클러스터링 되는 층)
        - 경쟁층은 사용자가 미리 정해놓은 군집의 수만큼 뉴런 수가 존재
- 입력층의 자료는 학습을 통하여 경쟁층에 정렬되는데, 이를 지도(map)라 부름
    - 입력 층에 있는 각각의 뉴런은 경쟁층에 있는 각각의 뉴런들과 연결되어 있으며, 이때 완전 연결(fully connected)되어 있음

코호넨 네트워크



각 학습 단계마다 입력층의 데이터 집합으로부터 하나의 표본 벡터(sample vector) x가 임의로 선택되었을 때, 프로토타입 벡터(prototype vector, 경쟁층의 각각의 뉴런을 의미)와의 거리를 유클리드 거리(euclidean distance)에 의해 계산하고 비교

- 입력 층의 표본 벡터에 가장 가까운 프로토타입 벡터를 선택하여 BMU(Best-Matching Unit)라 명명
- 그리고 코호넨의 승자 독점의 학습 규칙에 따라 BMU뿐만 아니라 위상학적 이웃(topological neighbors)에 대한 연결 강도를 조정

- 이처럼 SOM은 경쟁 학습으로 각각의 뉴런이 입력 벡터와 얼마나 가까운가를 계산하여 연결 강도(connection weight)를 반복적으로 재조정하여 학습
- 이와 같은 과정을 거치면서 연결강도는 입력 패턴과 가장 유사한 경쟁층 뉴런이 승자가 됨
- 결국 승자 독식 구조로 인해 경쟁층에는 승자 뉴런만이 나타나며, 승자와 유사한 연결 강도를 갖는 입력 패턴이 동일한 경쟁 뉴런으로 배열
- 따라서 SOM을 이용한 군집 분석은 역전파(back propagation) 알고리즘 등을 이용하는 인공신경망과 달리 단 하나의 전방 패스(feed-forward flow)를 사용함으로써 수행 속도가 매우 빠름
- 따라서 잠재적으로 실시간 학습처리를 할 수 있는 모델

✅ 요약: 쉽게 말하는 SOM(Self-Organizing Map)

SOM은 복잡한 데이터를 2차원 지도처럼 바꿔서, 비슷한 데이터끼리 가까운 위치에 정렬해주는 똑똑한 '지도 그리기 신경망'

- 입력값들 사이의 '비슷함'을 기준으로 가장 닮은 놈을 찾고, 그 주변까지 같이 조정해서 점점 정리된 지도를 만들어나가는 구조
    - 이때 지도 위의 점 하나하나는 '뉴런'이라고 부름

SOM(Self-Organizing Maps)이란?

비슷한 데이터를 가까이 배치하는 지도 정렬기

- SOM은 **비지도 학습(unsupervised learning)** 방식의 신경망이야.
- 어떤 데이터든 비슷한 특성끼리 **서로 가까운 위치**에 놓이도록 2차원 격자에 정렬하지.
- 예를 들어, 비슷한 성향의 고객을 지도 위에 배치하면, 서로 가까운 곳에 모이게 되는 거야.

어떻게 작동하냐면?

1. 두 개의 층으로 구성됨

- **입력층**: 사용자가 넣는 데이터 (예: 키, 나이, 성별 등 특징 벡터)

- **경쟁층(2차원 격자)**: 뉴런들이 쫙 깔린 지도처럼 생겼어. 여기에 데이터를 '배정'하게 됨.

2. 학습 순서 요약

- 하나의 **입력 벡터(데이터 샘플)**를 임의로 선택함.
- 모든 뉴런과 이 입력값 사이의 거리를 **유클리드 거리**로 계산.
- 가장 가까운 뉴런을 **BMU(Best Matching Unit)**라고 부름.
- 그 BMU 주변 이웃들까지 같이 '닮은 모습'으로 업데이트함.
- 이렇게 여러 번 반복하면, 점점 **데이터 구조를 반영한 지도**가 완성됨.

특징과 장점

고전 신경망과는 다른 방식

- **역전파(backpropagation)** 같은 계산 안 씀 → 훨씬 간단하고 빠름.
- 데이터 간의 **상대적 유사성**을 시각화하기 좋음.
- **실시간 처리**에도 적합해서 센서 네트워크 같은 데도 자주 쓰임.

쉬운 비유: SOM은 이런 거야!

> 데이터를 상자에 담는 게 아니라, 지도에 스티커 붙이듯 정리하는 거라고 보면 돼.
>
> 비슷한 성격의 친구들끼리 **옆자리에 앉게 배치**하는 교실 seating chart라고 보면 됨!

종료
