😀:제목 😎:소제목 😸👤 :내용 📜 📌📣⭐📖:책 💡 ✔️🤗 📑:ppt 📘📒 🔎: 검색

인공지능 기말고사 정리

😀DT

😎의사 결정 트리 (Decision Tree)

😸👤정의

- **귀납적 추론을 기반**으로 자료 집합을 적절한 분할 기준 또는 분할 테스트에 따라 부분 집합들로 나누는 과정을 **재귀적 반복 수행**하고 더이상 새로운 예측값이 추가되지 않거나 부분 집합의 노드가 목표변수와 같은 값을 지닐때 학습모델을 생성하는 알고리즘
- 주로 불연속 데이터를 다루며 노이즈가 발생해도 중단되거나 엉뚱한 결과를 보여주지 않고 강건

😸👤원리

- 훈련 데이터의 특성값/클래스 정보를 학습한 뒤 새롭게 주어진 테스트 데이터부터는 특성값으로 만 클래스를 예측하는것
    - 특성값(x값)
      → 고정된 집합의 특성들로 표현될 수 있어야한다.
    - 클래스(y값, target value)
      → 이산 출력값을 가져야한다.

⭐장점 & 단점 (📑5장 분류 26페이지)

- 장점
  - 데이터 분석결과가 나무(Tree) 구조로 표현되기 때문에 쉽게 이해하고 설명 가능
  - 분류 정확도가 비교적 높지 않게 평가되지만 결과에 대한 설명력이 높아 의사결정 시에 직접적으로 사용가능
- 단점
  - 데이터 특성이 특정 변수에 수직/수평적으로 구분되지 못할 때 분류율이 떨어지고, 트리가 복잡해지는 문제가 발생 가능
  - Hill Climbing 방식 및 Greedy 방식을 사용함으로 인해 최적의 해를 보장하지 못함
  - 데이터의 크기에 따라 정보값이 달라지고 결국 트리의 모양이 많이 달라질 수 있음

출처: https://ai-times.tistory.com/77 [ai-times]

😀:제목 😎:소제목 😺👤 :내용 📜 📌📣⭐📖:책 💡 ✔️🤗 📑:ppt 📘📒 🔎: 검색

😎엔트로피

😺👤정의

- 어떤 사건이 나한테 얼마만큼의 정보의 가치를 주는지 **확률적**으로 결정하는 함수이다.
- 확률의 **불확실성**을 수치로 나타냄

😺👤원리 (📑 5장 분류)

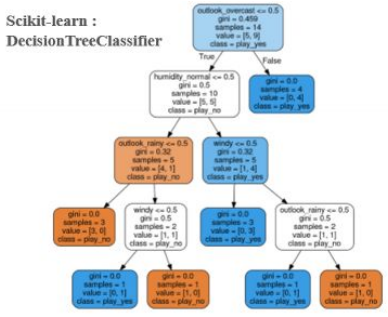
- $$E(S) := \sum_{i=1}^{c} p_i I(X_i) = \sum_{i=1}^{c} p_i \log_2 \frac{1}{p(X_i)} = \sum_{i=1}^{c} p_i \log_2 \frac{1}{p_i}$$

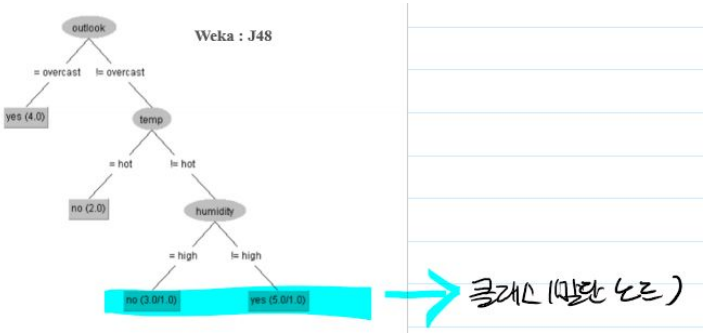
엔트로피 0 일때 = 출력은 매우 확실한 상태

😎의사결정 트리 생성 방법

1. 속성값마다 가지를 생성(각 가지에 속성값 하나씩 할당)
2. 속성값 별 엔트로피 $E(Sa) = 0$ **(출력이 매우 확실한 상태)**이면 leaf노드 생성
3. 속성값별 엔트로피 $E(Sa) \neq 0$ 이면 현재속성을 목표 속성으로 정하고 1~3의 과정을 반복

📑DT 의 예시





😀:제목 😎:소제목 😸👤 :내용 📜 📌📣⭐📖:책 💡 ✔️🤗 📑:ppt 📘📒 🔎: 검색

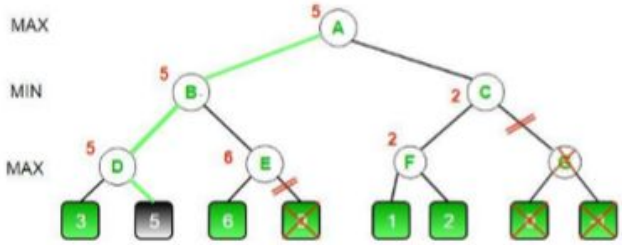
😎가지치기 (Pruning)

📜에러감소 프루닝 (📑 : **김영환**)

모든 노드 아래부분을 자르거나 결합한 뒤의 오류와 결합전의 오류를 비교하여 더이상 오류가 줄어들기 전까지 반복하는 단순하고 직관적인 방법.

📜룰 포스트 프루닝 (📑 5장 25페이지)

① 모든 학습 데이터를 가지고 ID3 알고리즘을 기반으로 의사결정 트리를 완성한다(오버피팅 발생 가능)
② 학습된 의사결정 트리를 룰 셋으로 변환 (룰: 루트부터 이파리까지의 경로)
③ 각 룰은 일련의 속성으로 구성되 있고, 정확도를 떨어뜨리는 속성을 제거한다.
④ 프루닝이 완성되면 정확도 순으로 정렬하고 이 순서대로 판별식에 활용한다.

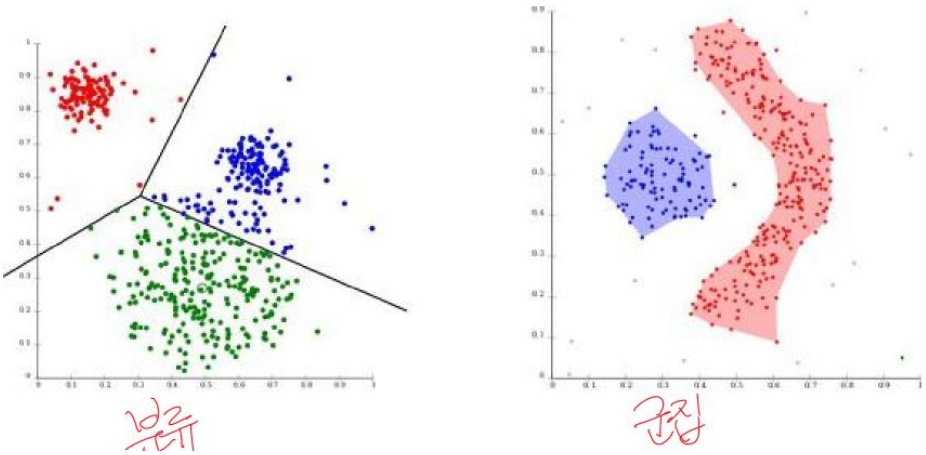


룰이라는 것은 루트 노드부터 잎 노드까지의 경로를 의미
트리를 모두 룰 형태로 변환 한 뒤 각 룰의 정확도를 구하고 정확도가 낮은 룰부터 제거하는 방법.

😀:제목 😎:소제목 😸👤 :내용 📜 📌📣⭐📖:책 💡 ✔️🤗 📑:ppt 📘📒 🔎: 검색

😀Clustering 군집

😸👤분류와 군집의 차이



분류	**지도학습** 사전에 정의된 그룹(범주)가 있는 (=레이블이 있는) 데이터를 대상으로 **예측 모델을 학습**하는 것
군집	**비지도 학습** 사전에 정의된 그룹(범주)가 없는(=레이블이 없는) 데이터를 대상으로 **최적화된 그룹이 되도록 묶는 것**

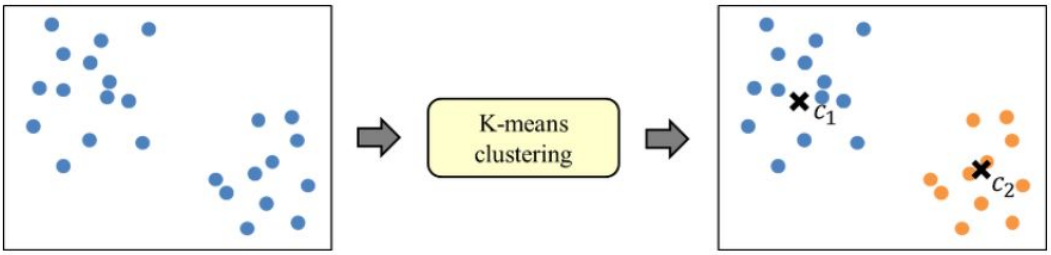
😀:제목 😎:소제목 😸👤 :내용 📜 📌📣⭐📖:책 💡 ✔️🤗 📑:ppt 📘📒 🔎: 검색

😎K-means 군집

😸👤정의

1. 알고리즘 정의

K-means clustering은 데이터를 입력받아 이를 소수의 그룹으로 묶는 알고리즘이다. 이 알고리즘은 아래의 [그림 1]처럼 label이 없는 데이터를 입력받아 각 데이터에 label을 할당함으로써 군집화를 수행한다. K-means clustering은 개념과 구현이 매우 간단한 기본적인 clustering 알고리즘이면서도 실행 속도가 빠르고, 특정한 형태의 데이터에 대해서는 매우 좋은 성능을 보여주기 때문에 많이 이용되고 있다.



[그림 1] K-means clustering의 동작

😸👤원리

K-means clustering은 벡터의 형태로 표현된 N개의 데이터 $X = \{x_1, x_2, \ldots, x_N\}$에 대하여 데이터가 속한 cluster의 중심과 데이터 간의 거리의 차이가 최소가 되도록 데이터들을 K개의 cluster $S = \{s_1, s_2, \ldots, s_K\}$에 할당한다. 많은 연구에서 K를 자동으로 설정하기 위한 시도가 이루어졌지만, 기본적으로 K는 데이터를 분석하고자 하는 사람이 직접 설정해주어야 한다. 어떠한 방법을 통해 K를 설정하였다고 가정할 때, k-means clustering의 동작은 아래의 최적화 문제로 표현될 수 있다.

알고리즘

① 클러스터 개수 결정(k=n)후 임의의 중심 n개 설정
② 모든 데이터는 n개의 중심까지 각각 거리를 계산한 후 가장 가까운 중심을 자신의 클러스터 중심이라고 정함
③ 각 클러스터마다 학습 데이터의 좌표값 평균을 계산한 후 이를 새로운 중심으로 설정
④ 새로 보정 후 이동된 중심을 기준으로 ②~③단계를 반복
⑤ 만약 모든 학습 데이터 중에서 자신이 속하는 클러스터를 변경하는 경우가 발생되지 않으면 학습 완료

😀:제목 😎:소제목 😸👤 :내용 📜 📌📣⭐📖:책 💡 ✔️🤗 📑:ppt 📘📒 🔎: 검색

📌 핵심 내용

- **K는 무엇인가?**
  - **중심점의 개수이다. (사람이 직접 지정한다.)**

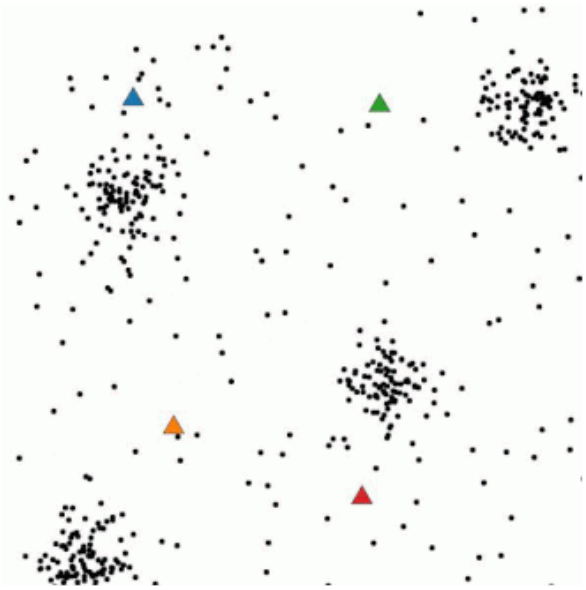
1. **초기에 임의의 중심점을K개를 지정한다.**
   - K(중심점의 수)를 자동으로 설정하기 위한 시도가 이루어졌지만, 기본적으로 K는 분석하고자 하는 사람이직접 설정해 주어야 한다.

   - **초기에 임의의 중심점 설정이 위험하지 않은가?**
     - 위험하다.

📑 **방지방법**
반복적으로 사용하여 가장 여러번 나타나는 군집을 사용 전체 데이터 중 일부만 샘플링하여 계층적 군집화를 수행한 뒤 초기 군집 중심 설정 사전 데이터 분포의 정보를 사용하여 초기 중심 설정 하지만 많은 경우 초기 중심 설정이 최종 결과에 큰 영향을 미치지는 않음

2. **각각의 K개의 중심점으로부터 데이터들의 거리를 계산한다.**
   - 그후 각 데이터들로부터 가장 가까운 중심점을 각 데이터들의 군집점으로 사용한다.
3. **각 중심점을 자신에게 속한 데이터들의 중심 좌표로 이동시킨다.**

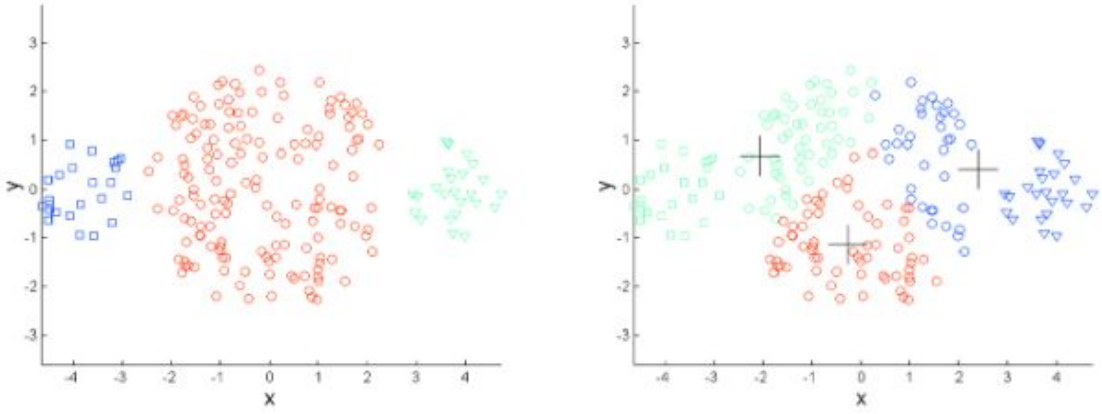
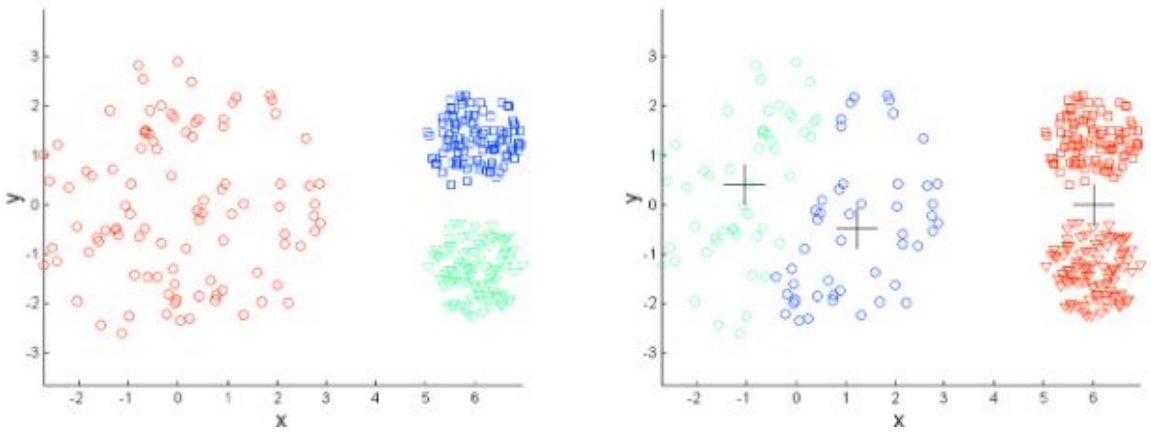
4. **다시 2번부터 반복적으로 수행한다.**



😀:제목 😎:소제목 😸👤 :내용 📜 📌📣⭐📖:책 💡 ✔️🤗 📑:ppt 📘📒 🔎: 검색

⭐장점 & 단점

📜 장점 & 단점(문제점)

단점
📑 군집의 크기가 다른 경우 군집을 잘 찾지 못한다. 군집의 크기가 다를 경우 제대로 작동하지 않을 수 있습니다. 
📑 군집의 밀도가 다른 경우 군집을 잘 찾지 못한다. 군집의 밀도가 다를 경우에도 마찬가지입니다. 

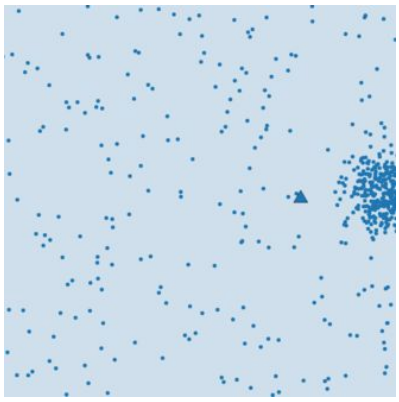
😀:제목 😎:소제목 😸👤 :내용 📜 📌📣⭐📖:책 💡 ✔️🤗 📑:ppt 📘📒 🔎: 검색

🔎 오목한 형태의 군집 모델 특성 구별에 어려움

🔎 결과해석의 어려움

- 탐색적인 분석방법으로의 장점을 가지고 있는 반면에 사전에 주어진 목적이 없으므로 결과를 해석하는데 있어서 어려움 존재

🔎 아웃라이어, 노이즈에 민감

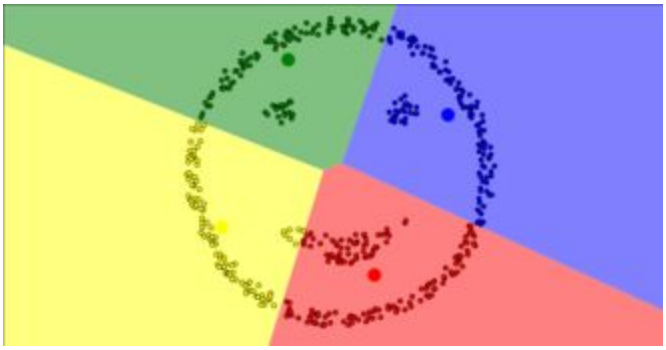


실제 데이터 클러스터는 오른쪽 중간에 위치해있는데 반해 클러스터의 중심은 클러스터로부터 왼쪽으로 상당히 떨어진 지점에 위치한다. 이는 실제 클러스터 외부에 퍼져 있는 **이상값들이 중심점 계산에 있어 그 값을 크게 왜곡시켰기 때문이다.** 이를 방지하기 위해 k-평균 알고리즘을 실시하기 전에 이상값을 제거하는 프로세스를 먼저 실행하거나, 분할법의 일종인 k-대푯값 알고리즘 (k-medoids algorithm) 을 이용하면 이상값의 영향을 줄일 수 있다.

🔎 초기 중심 선정에 따라 글로벌 최소값에 도달하지 못하는 경우 발생

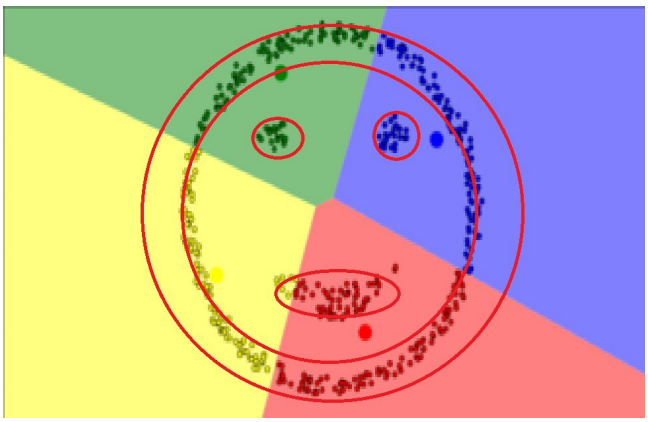
🔎 구형 (spherical) 이 아닌 클러스터를 찾는 데에는 적절하지 않다.
📑 (지역적 패턴이 존재하는 군집을 판별하기 어려움)

k-평균 알고리즘은 비용 함수를 계산할 때 중심점과 각 데이터 오브젝트간의 거리를 계산하게 된다. 이 때 사용하는 거리는 유클리드 거리이다. 따라서, 알고리즘 수행 시 중심점으로부터 구형으로 군집화가 이루어지게 된다. **만약 주어진 데이터 집합의 분포가 구형이 아닐 경우에는 클러스터링 결과가 예상과 다를 수 있다.** 아래의 경우를 살펴보자.



비구형 데이터군집 (틀린 분류)

😀:제목 😎:소제목 😸👤 :내용 📜 📌📣⭐📖:책 💡 ✔️🤗 📑:ppt 📘📒 🔎: 검색



정상 분류

🔎 장점
주어진 자료에 대한 사전정보 없이 의미 있는 자료구조를 찾아 낼 수 있다.
간단한 알고리즘으로 대규모에도 적용이 가능하다. (계산시간이 짧다)
탐색적인 기법 ● 군집분석은 그 자체가 대용량 데이터에 대한 탐색적 인 기법으로서, 주어진 데이터의 내부구조에 대한 사 전적인 정보 없이 의미 있는 자료구조를 찾아낼 수 있는 방법
다양한 형태의 데이터에 적용 가능 ● 분석을 위해서는 기본적으로 관찰치 간의 거리를 데 이터형태에 맞게만 정의하면, 거의 모든 형태의 데이 터에 대하여 적용이 가능한 방법
분석방법의 적용 용이성 ● 대부분의 군집방법이 분석대상 데이터에 대해 사전정 보를 거의 요구하지 않음 적용 유리 즉, 모형화를 위한 분석과 같이 사전에 특정 변수에 대한 역할 정의가 필요하지 않고 다만 관찰치 들 사 이의 거리만이 분석에 필요한 입력자료로 사용.
볼록한 형태의 데이터셋에 잘 대응

😀:제목 😎:소제목 😸👤 :내용 📜 📌📣⭐📖:책 💡 ✔️🤗 📑:ppt 📘📒 🔎: 검색

😸👤최적의 군집 수 결정

Sum of Squared Error (SSE)

- 모든 군집들에 대해 중심값과 개체들과의 거리의 합을 구하고, 최종 가장 작은 값을 취함

$$SSE = \sum_{i=1}^{K} \sum_{x \in C_i} dist(x, c_i)^2$$

Silhouette 통계량

- 관측시와 타 군집 내의 개체와의 거리의 합에서 관측치와 현재 군집 내의 개체와의 거리 합과의 차

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}}, \quad \bar{S} = \frac{1}{n} \sum_{i=1}^{n} S(i)$$

😀:제목 😎:소제목 😸👤 :내용 📜 📌📣⭐📖:책 💡 ✔️🤗 📑:ppt 📘📒 🔎: 검색

😎K-medoids 군집

😸👤정의

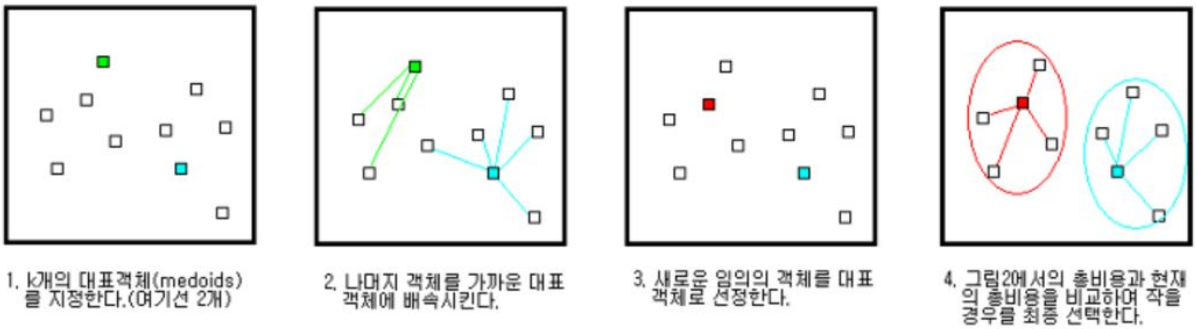
- Medoids = 중앙객체(대표객체)

🔎 위키백과
k-평균 알고리즘(K-means algorithm)은 주어진 데이터를 k개의 클러스터로 묶는 알고리즘으로, 각 클러스터와 거리 차이의 **분산을 최소화하는 방식으로 동작**한다. 이 알고리즘은 자율 학습의 일종으로, 레이블이 달려 있지 않은 입력 데이터에 레이블을 달아주는 역할을 수행한다.

😸👤원리

📑 알고리즘

1. n개의 객체 중 **대표 객체**(medoids)를 k개 지정한다. (n>k)
2. k개의 medoids를 지정 후, 나머지 객체를 유사성이 가장 높은 medoid에 배속한다.
   여기서 **유사성은 거리측도를 활용**한다.
3. medoid가 아닌 다른 객체를 임의로 지정한다.
4. 본래의 medoid와 임의의 medoid간의 총 비용(cost)를 계산한다.
   (cost는 거리측도의 값을 모두 더한 것을 사용한다.)
5. 만약 총 비용이 음수인 경우, 임의의 medoid를 기존의 medoid와 교체한다.
6. 변화가 없을 때까지 2~5를 반복한다.



😀:제목 😎:소제목 😸👤 :내용 📜 📌📣⭐📖:책 💡 ✔️🤗 📑:ppt 📘📒 🔎: 검색

📌핵심내용

🔎 참고자료

k-means는 평균값을 구하는 연산을 수행하기 때문에 잡음이나 이상치(아웃라이어)에 민감하다. 이러한 단점을 해결하기 위해 나온것이 k-medoids 알고리즘이다. k-medoids는 클러스터의 대표값으로 오브젝트의 중심점을 구하는 것이 아니라, 오브젝트 중에서 클러스터를 대표할 수 있는 가장 가까운 대표 오브젝트를 뽑는다.

k-means의 경우 평균을 대표값으로 가져가기 때문에 분산을 기준으로 알고리즘이 진행되는것에 반해 **k-medoid는 중앙값을 대표값**으로 가져가므로 아래 처럼 절대오차를 기준으로 알고리즘이 진행되게 된다.

$$E = \sum_{i=1}^{k} \sum_{p \in C_i} dist(p, o_i),$$

💡 **K-means 와 K-medoids의 비교**

중앙값은 평균에 비해 이상치에 영향을 덜 받기 때문에 k-medoid가 k-means보다 안정적이다. 그러나 k-medoid 알고리즘의 경우 **kmeans에 비해 계산복잡도가 훨씬 높기 때문에 훨씬 느리며 이는 비용과 직결된다.** 즉 작은 데이터에서는 잘 작동하나 대규모 데이터를 다룰때는 적절하지 않다는것이다.

⭐장점 & 단점 (K-means vs. K-medoids)

📑 장. 단점 (PPT 내용)

K-means

- 장점
  - 개념이 명확하고, 직관적임
  - 학습을 위한 계산이 빠름
  - $O(n)$의 계산량
  - 볼록한 형태의 데이터셋에 잘 대응
- 단점
  - 오목한 형태의 군집 모델 특성 구별에 어려움
  - 아웃라이어, 노이즈에 민감
  - 초기 중심 선정에 따라 글로벌 최소값에 도달하지 못하는 경우 발생

K-medoids

- 장점
  - 실제 데이터세트에 있는 값을 중심점으로 하기에 아웃라이어, 노이즈 처리가 우수
  - 강건하고 수렴
- 단점
  - 오목한 형태의 군집 모델 특성 구별에 어려움
  - 새로운 중심을 데이터 중 하나로 선정하므로 k-means에 비해 계산량이 많음
  - $O(k \cdot (n-k)^2 \cdot t)$의 계산량
  - 초기 중심 선정에 따라 글로벌 최소값에 도달하지 못하는 경우 발생

- K-means와는 다르게 임의의 중심점이 아닌 N개의 객체 중 **대표 객체**(medoids)를 K개 지정한다.(K<N)
- 여기서 중심으로 사용되는 대표객체와 나머지 객체간의 유사성은 K-means와 동일하게 거리의 척도를 활용한다.

😀:제목 😎:소제목 😸👤 :내용 📜 📌📣⭐📖:책 💡 ✔️🤗 📑:ppt 📘📒 🔎: 검색

😎DBSCAN 군집

😸👤정의

- DBSCAN : Density-Based Spatial Clustering of Applications with Noise
- **밀도 기반 클러스터링**
- 점이 세밀하게 몰려 있어서 밀도가 높은 부분을 클러스터링 하는 방식이다.
- 어느점을 기준으로 반경 x내에 점이 n개 이상 있으면 하나의 군집으로 인식하는 방식이다.

노이즈, 아웃라이어에 강건한 군집 모델

"밀도 있게 연결되 있는(connected) 데이터 집합은 동일한 클러스터"

- 일정한 밀도를 가지는 데이터의 무리가 마치 체인처럼 연결되 있으면 거리 개념과 상관없이 같은 클러스터로 판단함

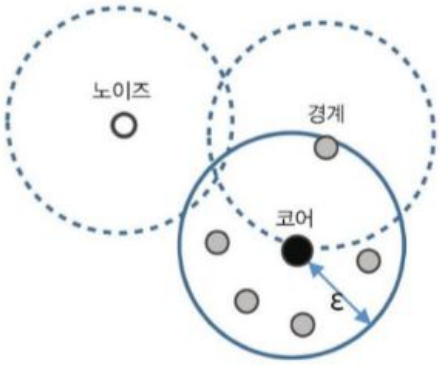


그림 6.2 DBSCAN의 3가지 데이터 형태

DBSCAN 용어 정리

X := 학습 데이터 전체 집합

ε := 밀도 측정 반지름

MinPts := 반지름 ε 내에 있는 최소 데이터 개수

N(x) := 데이터 x의 반지름 ε 내에 있는 이웃 데이터(neighbor) 수

{x} := 데이터 x의 반지름 ε 내에 있는 이웃 데이터

- $x := x_{core}$: 만약 $N(x) \geq$ MinPts $\forall_{x \in X}$
- $x := x_{border}$: 만약 $x \in \{x_{core}\}$이고 $N(x) <$ MinPts $\forall_{x \in X}$
- $x := x_{noise}$: 만약 $x \notin \{x_{core}\}$이고 $N(x) <$ MinPts $\forall_{x \in X}$

📑 ppt 사진 참고

🔎 **참고자료**
k-평균 군집이나 계층적 군집 알고리즘의 경우 데이터 간의 거리를 이용하여 클러스터를 나누는데에 반해, DBSCAN 알고리즘은 데이터 포인트가 세밀하게 몰려 있어 밀도가 높은 부분을 군집화하는 방식이다.

😀:제목 😎:소제목 😸👤 :내용 📜 📌📣⭐📖:책 💡 ✔️🤗 📑:ppt 📘📒 🔎: 검색

😸👤원리

DBSCAN은 밀도를 기반으로 하여 군집화하는 매우 유용한 군집 알고리즘이다. k-평균 군집이나 계층적 군집 알고리즘의 경우 데이터 간의 거리를 이용하여 클러스터를 나누는데에 반해, DBSCAN 알고리즘은 데이터 포인트가 세밀하게 몰려 있어 밀도가 높은 부분을 군집화하는 방식이다.

- 먼저 DBSCAN 알고리즘은 특성 공간(Feature Space)에서 데이터가 밀집해있는 지역의 포인트를 찾는다.
  이러한 지역을 특성 공간의 **밀집 지역(Dense Region)**이라 한다.

  이러한 데이터의 밀집 지역이 하나의 클러스터를 구성하며, 비교적 비어있는 지역을 경계로 다른 클러스터와 구분하는 것이다.

- 밀집 지역에 있는 포인트를 **핵심 샘플(Core Point)**라고 하며, 임의로 선정한 하나의 데이터 포인트에서 **지정 거리(eps)** 안에 데이터가 **최소 샘플 개수(min_samples)**만큼 들어 있으면 이 데이터 포인트를 핵심 샘플로 분류한다. 따라서 지정해 준 거리보다 가까이 있는 핵심 샘플은 DBSCAN 알고리즘에 의해 동일한 클러스터로 합쳐진다.

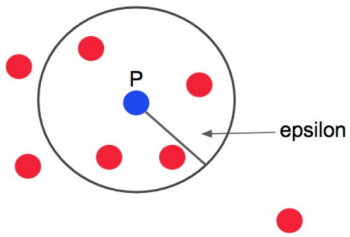
  간단하게 말해서, 어느 포인트를 기준으로 지정 반경 내에 데이터 포인트가 n개 이상 있으면 하나의 클러스터로 인식하는 방식이다.

😀:제목 😎:소제목 😸👤 :내용 📜 📌📣⭐📖:책 💡 ✔️🤗 📑:ppt 📘📒 🔎: 검색

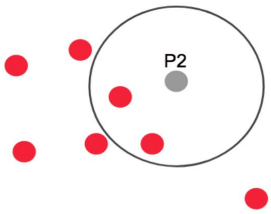
🔎 [참고자료]

최종적으로 데이터는 핵심(코어) 포인트, 경계 포인트(클러스터 내에 속하나 지정 반경 내 최소 샘플 개수를 만족하지 못한 포인트), 잡음 포인트 세 가지로 나뉜다.

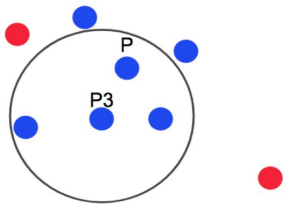
아래 그림에서 minPts = 4 라고 하면, 파란점 P를 중심으로 반경 epsilon 내에 점이 4개 이상 있으면 하나의 군집으로 판단할 수 있는데, 아래 그림은 점이 5개가 있기 때문에 하나의 군집으로 판단이 되고, P는 core point가 된다.



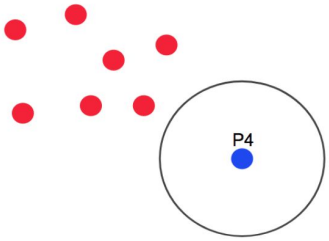
아래 그림에서 회색점 P2의 경우 점 P2를 기반으로 epsilon 반경내의 점이 3개 이기 때문에, minPts=4에 미치지 못하기 때문에, 군집의 중심이 되는 core point는 되지 못하지만, 앞의 점 P를 core point로 하는 군집에는 속하기 때문에 이를 boder point (경계점)이라고 한다.



그런데 P3를 중심으로 하는 반경내에 다른 core point P가 포함이 되어 있는데, 이 경우 core point P와 P3는 연결되어 있다고 하고 하나의 군집으로 묶이게 된다.



마지막으로 아래 그림의 P4는 어떤 점을 중심으로 하더라도 minPts=4를 만족하는 범위에 포함이 되지 않는다. 즉 어느 군집에도 속하지 않는 outlier가 되는데, 이를 noise point라고 한다.



😀:제목 😎:소제목 😸👤 :내용 📜 📌📣⭐📖:책 💡 ✔️🤗 📑:ppt 📘📒 🔎: 검색

📌핵심내용

🔎참고자료
DBSCAN 알고리즘의 매개변수는 **최소 샘플 개수(min_samples)와 지정 거리(eps)**가 있다. **지정 거리가 증가**하면 하나의 클러스터에 더 많은 포인트가 포함된다. 이는 **클러스터를 커지게 하지만, 여러 클러스터를 하나로 합치게도 만든다.** 일반적으로 지정 거리가 가까운 포인트의 범위를 결정하기 때문에 매우 중요하다. **지정 거리를 매우 작게** 하면 **어떤 포인트도 핵심 포인트가 되지 못하고, 모든 포인트가 잡음 포인트가 될 수 있다.** 반대로 지정 거리를 매우 크게 하면 모든 포인트가 단 하나의 클러스터에 속하게 된다. 최소 샘플 개수 설정은 조밀하지 못한 지역에 있는 포인트들이 잡음 포인트가 될 것인지 하나의 클러스터가 될 것인지를 결정하는 데 중요한 역할을 한다.

😀:제목 😎:소제목 😸👤 :내용 📜 📌📣⭐📖:책 💡 ✔️🤗 📑:ppt 📘📒 🔎: 검색

⭐장점 & 단점 (📑 : 6장 군집 11페이지)

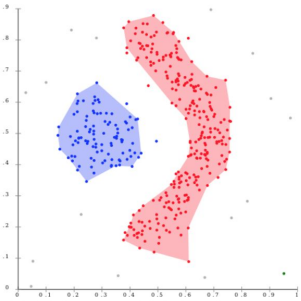
✔️장점

$O(n \cdot log(n))$의 계산량

다양한 기하학적 형태의 군집 유형을 구분해 낼 수 있음.

군집의 개수를 자동으로 찾아줌

노이즈를 구분하여 버림으로 의미없는 군집 생성을 차단함

🔎 장점
• **K Means와 같이 클러스터의 수를 정하지 않아도 되며,** • **클러스터의 밀도에 따라서 클러스터를 서로 연결하기 때문에 기하학적인 모양을 갖는 군집도 잘 찾을 수 있으며**  • **Noise point를 통하여, outlier 검출이 가능하다**

✔️단점

밀도 반지름(ε) 및 최소 이웃 수(MinPts)가 문제의 특성에 따라 민감하게 작용

데이터의 밀도가 다양한 데이터에 적합하지 않음

- 어떤 군집은 밀도가 듬성듬성한데, 어떤 군집은 오밀조밀 할 때

ε 거리를 추정하기 어려움

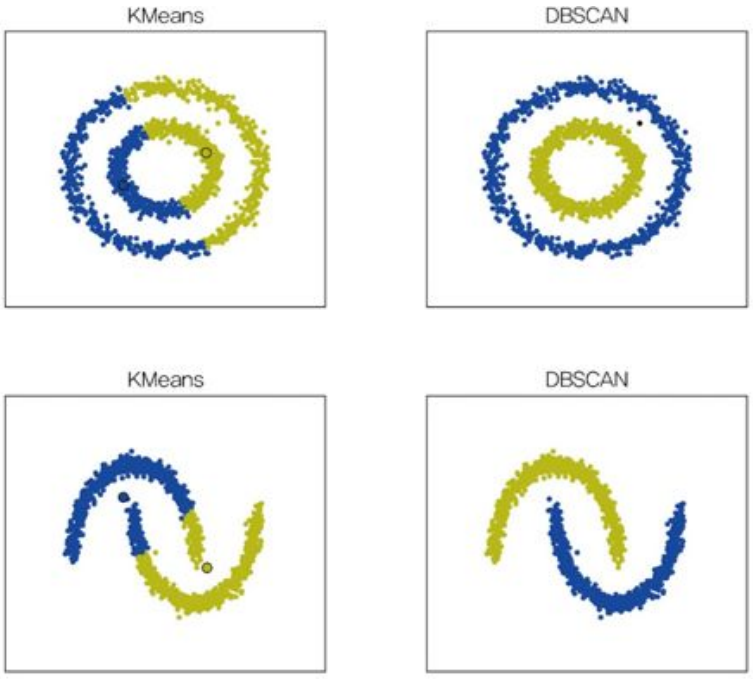
위의 장점들이 단점이 될 수도 있음

- 기하학적 군집 구분 -> 군집의 중심과 특성을 정하기가 어려움.
- 군집의 개수를 3개로 만들고 싶은데 딱 3개로 나오지 않을 수 있음. (거리와 최소개수를 변경해가면서 찾아야 함)
- 노이즈를 구분 -> "군집에 포함되지 않은 개체는 어떻게 처리할 것 인가?"에 대한 문제

😀:제목 😎:소제목 😸👤 :내용 📜 📌📣⭐📖:책 💡 ✔️🤗 📑:ppt 📘📒 🔎: 검색

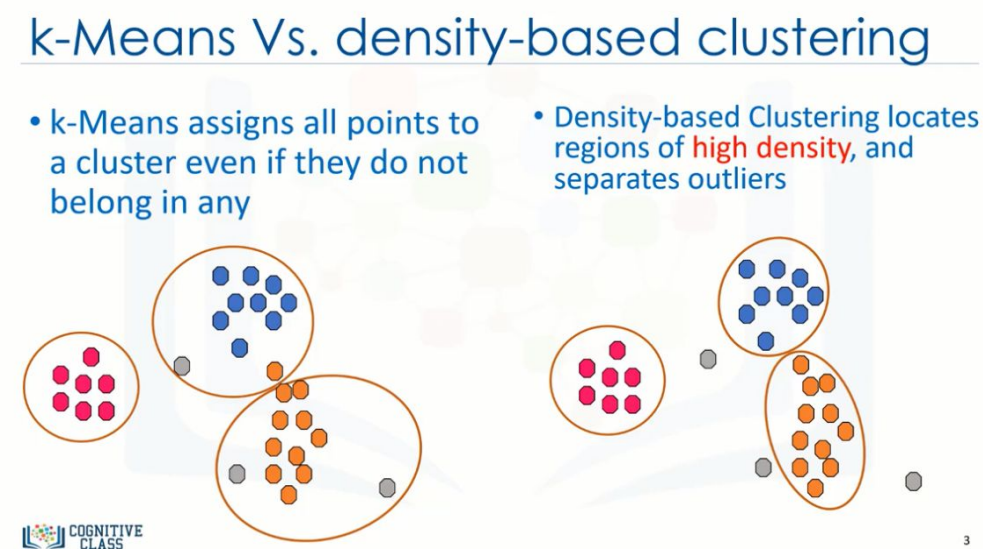
📜k-Means와 DBSCAN의 차이

📑 ppt 내용



영상 https://youtu.be/6jl9KkmgDIw

🔎 검색내용



회색의 버리는 것들이 생김

😀:제목 😎:소제목 😸👤 :내용 📜 📌📣⭐📖:책 💡 ✔️🤗 📑:ppt 📘📒 🔎: 검색

😎계층형 군집

😸👤정의

🔎 개념

Hierarchical clustering (한글 : 계층적 군집 분석) 은 비슷한 군집끼리 묶어 가면서 최종적으로는 하나의 케이스가 될때까지 군집을 묶는 클러스터링 알고리즘이다.

군집간의 거리를 기반으로 클러스터링을 하는 알고리즘이며, K Means와는 다르게 군집의 수를 미리 정해주지 않아도 된다.

참고자료

📑 개념

개념
- 거리 측정 기반 유사한 특성을 지닌 데이터를 greedy하게 묶어 이진 트리 형태로 만들어 가는 방법

Advantages:
- 사전에 클러스터 개수를 알 필요 없음
- 구현이 쉽고, 대부분의 경우에서 좋은 성능을 나타냄

Disadvantages:
- $O(n^2 \cdot log(n))$의 계산량
- 이전에 수행된 결과를 되돌릴 수 없음
- 거대한 클러스터들을 나누는 작업은 어려움
- 노이즈와 아웃라이어 민감
- 많은 양의 클러스터의 경우, 덴드로그램에서 인식하기가 어려움

Applications
- Recommendation engines
- Social media analysis
- Search result grouping
- Image segmentation

- Greedy : 탐욕스러운

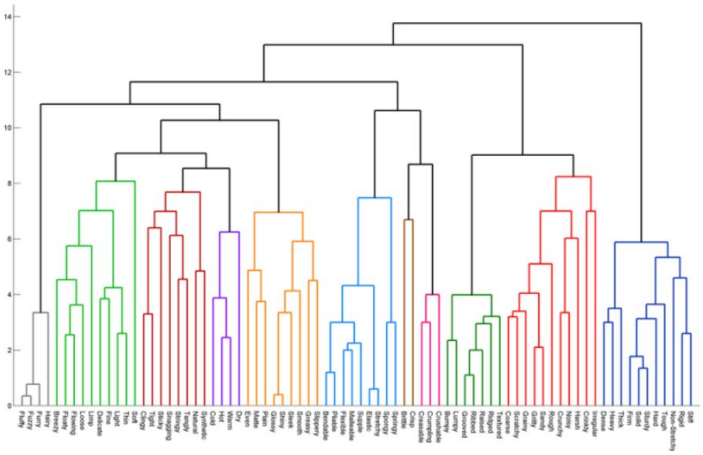
😀:제목 😎:소제목 😸👤 :내용 📜 📌📣⭐📖:책 💡 ✔️🤗 📑:ppt 📘📒 🔎: 검색

😸👤원리

🔎 **참고자료**

HC란 계층적 트리 모형을 이용해 개별 개체들을 순차적, 계층적으로 유사한 개체 내지 그룹과 통합하여 군집화를 수행하는 알고리즘입니다. K-평균 군집화(K-means Clustering)와 달리 군집 수를 사전에 정하지 않아도 학습을 수행할 수 있습니다.

개체들이 결합되는 순서를 나타내는 트리형태의 구조인 **덴드로그램(Dendrogram)** 덕분입니다. 아래 그림과 같은 덴드로그램을 생성한 후 적절한 수준에서 트리를 자르면 전체 데이터를 몇 개 군집으로 나눌 수 있게 됩니다.



1) 전체 데이터의 유사성 행렬을 구한다.

- 각 데이터를 하나의 군집으로 간주한다.
  - **유사성 행렬 계산** 이란
    데이터간의 거리가 최소가 되는 두 데이터를 하나의 군집으로 묶는것

2) repeat (반복)

- 가장 가까운 두 군집을 묶어 한 군집으로 한다.
- 새롭게 형성된 군집을 포함한 전체 군집간의 유사성 행렬을 구한다.

3) until (동안, ~까지 반복)

- (군집의 수가 하나가 될 때까지)

😀:제목 😎:소제목 😸👤 :내용 📜 📌📣⭐📖:책 💡 ✔️🤗 📑:ppt 📘📒 🔎: 검색

📌핵심내용

🔎 참고자료

- **두가지를 결정해야 한다.**
  1. 두 데이터 포인트의 비슷하지 않은 정도, **즉 거리(dissimilarity) 정의**
     - 다른 거리를 쓸 수 있지만, 보통 유클리드 거리(Euclidean distance)가 쓰인다.
  2. 두 데이터 포인트의 **그룹(cluster)끼리의 거리(dissimilarity)를 정의**
     - 두 점 사이의 거리가 아닌, 두 클러스터 사이의 거리를 정의해야 한다.
     - 거리(dissimilarity)의 일반화된 개념으로 linkage라고 부른다.

📜 클러스터간 거리척도

단일 연결(single-linkage) : 이종 클러스터 간 최소값을 갖는 데이터의 거리 계산

평균 연결(average-linkage) : 이종 클러스터 간 모든 데이터를 대상으로 평균 계산

완전 연결(complete-linkage) : 이종 클러스터 간 최대값을 갖는 데이터의 거리 계산

중심법(centroid method) : 이종 클러스터 각각의 중심을 구하고, 이종 클러스터의 중심 간의 거리를 계산

워드법(ward's method) : Distance between two clusters, A and B, is how much the sum of squares will increase when they are merged

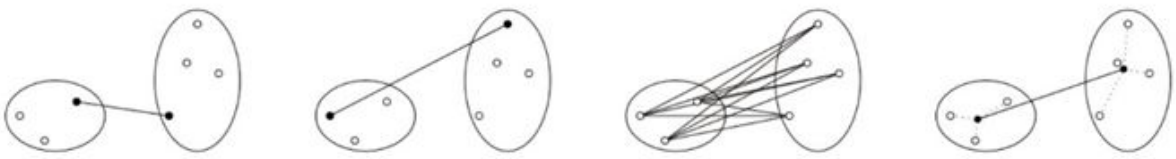


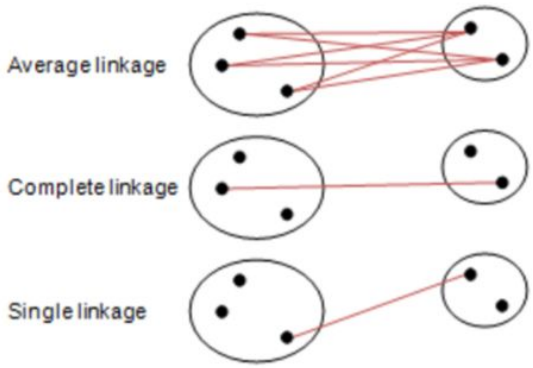
그림 6.6 병합적 계층형 군집에서 유사성 계산 방법

😀:제목 😎:소제목 😸👤 :내용 📜 📌📣⭐📖:책 💡 ✔️🤗 📑:ppt 📘📒 🔎: 검색

🔎 검색자료

계층 분석 방식

앞의 코드에서, linkage 함수에서 method 를 사용했다. 이에 대해서 알아보자.
Hierachical clustering의 기본 원리는 두 클러스터 사이의 거리를 측정해서 거리가 가까운 클러스터끼리 묶는 방식이다.
그러면 두 클러스터의 거리를 측정할때 어디를 기준점으로 할것인가를 결정해야 하는데 다음 그림을 보자.



출처 : https://www.multid.se/genex/onlinehelp/hs515.htm

앞의 코드에서 사용한 complete linkage 방식은 두 클러스터상에서 가장 먼 거리를 이용해서 측정하는 방식이고 반대로 single linkage 방식은 두 클러스터에서 가장 가까운 거리를 사용하는 방식이다.
average linkage 방식은 각 클러스터내의 각 점에서 다른 클러스터내의 모든 점사이의 거리에 대한 평균을 사용하는 방식이다.

이 linkage 방식에 따라서 군집이 되는 모양이 다르기 때문에, 데이타의 분포에 따라서 적절한 linkage 방식을 변화 시켜가면서 적용해가는 것이 좋다.

■ 최장 연결법(Complete Linkage Method)

최장 거리는 다음과 같이 정의합니다.

$$d(U, V) = \max[d(x, y) | x \in U, y \in V]$$

두 군집 U와 V사이의 거리 d_{UV}를 각 군집에 속하는 임의의 두 개체들 사이의 거리 중 최장거리로 정의하여 가장 유사성이 큰 군집을 묶어 나가는 방법입니다. 이와 같은 최장 연결법은 앞의 최단 연결법과는 대조적인 관계가 있습니다. 일반적으로, 최단 연결법이 고립된 군집을 찾는데 유용하다면, 최장 연결법은 군집들의 응집성에 중점을 둔다고 하겠습니다. 이런 점에 대한 보완 방법으로써 중심 연결법, 중위수 연결법, 평균 연결법 등이 제안되었습니다.

😀:제목 😎:소제목 😸👤 :내용 📜 📌📣⭐📖:책 💡 ✔️🤗 📑:ppt 📘📒 🔎: 검색

■ 평균 연결법(Average Linkage Method)

크기가 각각 N_1, N_2인 두 군집 U, V사이의 거리를, 각 군집에서 하나씩의 개체를 택해 연결한 모든 가능한 $N_1 \times N_2$가지의 거리 d_{ij} 의 평균을 다음과 같이 정의합니다.

$$d(U, V) = 1/(N_1 N_2) \sum_i \sum_j d_{ij}$$

두 군집 U와 V 사이의 거리 d_{UV} 를 각 군집에 속하는 모든 개체들의 평균 거리로 정의하여 가장 유사성이 큰 군집을 묶어 나가는 방법입니다.

■ 중심 연결법(Centroid Linkage Method)

U의 평균을 $\overline{X}_i$ 라고 표기하고 P를 두 군집 사이의 유클리드 거리와 같은 비 상사성척도라 하면 두 군집 U, V 사이의 거리는 다음과 같이 정의됩니다.

$$d(U, V) = P(\overline{X}_1, \overline{X}_2)$$

두 군집 U의 중심점과 군집 V의 중심점 사이의 거리를 두 군집 사이의 거리로 정의하여 가장 유사성이 큰 군집을 묶어 나가는 방법입니다.

■ 중위수 연결법(Median Linkage Method)

두 군집 U와 V사이의 거리 d_{UV} 를 각 군집에 속하는 임의의 두 개체들 평균을 합하여 2로 나눈 값(군집의 크기를 고려하지 않은 단순 평균)을 근간으로 정의하여 가장 유사성이 큰 군집을 묶어 나가는 방법입니다.

😀:제목 😎:소제목 😸👤 :내용 📜 📌📣⭐📖:책 💡 ✔️🤗 📑:ppt 📘📒 🔎: 검색

📜 병합형/ 분활형 군집

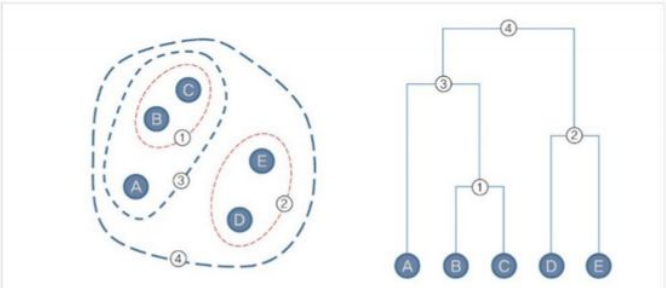
Agglomerative clustering 예제

상향식 군집

개체들을 가까운 집단부터 차례로 묶어 나가는 방식, dendrogram 생성

알고리즘

① 모든 데이터를 단일 클러스터로 정의
② 각 클러스터 간 유사성을 계산
③ 유사성이 가장 높은 두 개의 클러스터를 합침
④ ②~③단계를 (전체 클러스터 개수 == 1)이 충족 시까지 반복



Divisive clustering

하향식 군집

전체 데이터의 영역을 특정 기준에 의해 동시에 구분

각 개체들은 사전 정의된 군집 개수 중 하나에 속하게 됨

알고리즘

① 모든 데이터를 포함하고 있는 단일 클러스터 정의
② 각 클러스터 간 유사성을 계산
③ 유사성이 가장 낮은 두 개의 클러스터를 분리
④ ②~③단계를 (전체 클러스터 개수 == 데이터 갯수)이 충족 시까지 반복

😀:제목 😎:소제목 😺👤 :내용 📜 📌📣⭐📖:책 💡 ✔️🤗 📑:ppt 📘📒 🔎: 검색

⭐장점 & 단점

✔️장점 (📑)

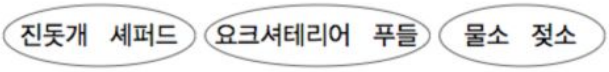
사전에 클러스터 개수를 알 필요 없음
구현이 쉽고, 대부분의 경우에서 좋은 성능을 나타냄

✔️단점 (📑)

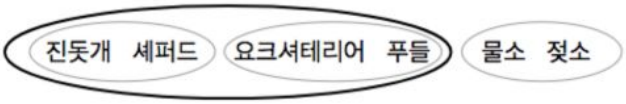
$O(n^2 \cdot log(n))$의 계산량
이전에 수행된 결과를 되돌릴 수 없음
거대한 클러스터들을 나누는 작업은 어려움
노이즈와 아웃라이어 민감
많은 양의 클러스터의 경우, 덴드로그램에서 인식하기가 어려움

😺👤예시 (🔎)

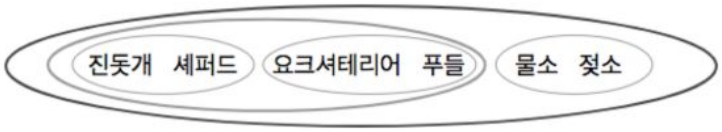
예를 들어서 설명해보자
"진돗개,세퍼드,요크셔테리어,푸들, 물소, 젖소" 를 계층적 군집 분석을 하게 되면
첫번째는 중형견, 소형견, 소와 같은 군집으로 3개의 군집으로 묶일 수 있다.



이를 한번 더 군집화 하게 되면 [진돗개,셰퍼드] 와 [요크셔테리어,푸들] 군집은 하나의 군집(개)로 묶일 수 있다.



마지막으로 한번 더 군집화를 하게 되면 전체가 한군집(동물)으로 묶이게 된다.



😀:제목 😎:소제목 😸👤 :내용 📜 📌📣⭐📖:책 💡 ✔️🤗 📑:ppt 📘📒 🔎: 검색

📑 군집 정리

- K-means
  - 대표적인 분리형 군집화 알고리즘
  - 초기 중심을 임의의 k개로 설정
  - 각 개체는 가장 인접한 클러스터 중심에 할당되며, 같은 중심에 할당된 개체들이 모여 하나의 군집을 형성

$$X = C_1 \cup C_2 \cdots \cup C_k,\ C_i \cap C_j = \emptyset,\ i \neq j$$

$$\underset{c}{\operatorname{argmin}} \sum_{i=1}^{K} \sum_{x_j \in C_i} \|x_j - c_i\|^2$$

- DBSCAN
  - 일정한 밀도를 가지는 데이터의 무리가 마치 체인처럼 연결되 있으면 거리 개념과 상관없이 같은 클러스터로 판단함

- Hierarchical clustering
  - 거리 측정 기반 유사한 특성을 지닌 데이터를 greedy하게 묶어 이진 트리 형태로 만들어 가는 방법
  - 사전에 클러스터 개수를 알 필요 없음
  - 구현이 쉽고, 대부분의 경우에서 좋은 성능을 나타냄

😀:제목 😎:소제목 😸👤 :내용 📜 📌📣⭐📖:책 💡 ✔️🤗 📑:ppt 📘📒 🔎: 검색

😀강화학습

😸👤정의

🔎 **참고자료**
강화 학습(Reinforcement learning)은 기계 학습의 한 영역이다. 행동심리학에서 영감을 받았으며, 어떤 환경 안에서 정의된 에이전트가 현재의 상태를 인식하여, 선택 가능한 행동들 중 보상을 최대화하는 행동 혹은 행동 순서를 선택하는 방법이다.

🔎 **에드워드 손다이크**
“인간이 가진 지능, 성격, 기술은 *끊임없는 시행착오를 거쳐 습득되어진다.”* **효과의 법칙:**

😸👤원리

⭐장점 & 단점