군집분석(Clustering)

개체들을 분류하기 위한 기준이 없는 상태에서 주어진 데이터의 속성값들을 고려해 유사한 개체끼리 그룹(클러스터)화하는 방법

그룹내 차이를 줄이고 그룹간 차이는 최대화 하도록 하여 대표성을 찾는 원리로 구현되는 것이 일반적.

클러스터링은 원래 기계학습의 비지도학습(Unsupervised Learning)의 기법 중 하나로 분류할 수 있는데 세그멘테이션 기법의 일종이라고도 볼 수 있어 모든 학술 분야에서 두루 다루며 데이터마이닝 분야에서 용도가 더 많아 데이터마이닝으로 분류하였다.

비계층적 군집분석 (Non-Hierarchical Clustering)

* 중심 기반(Center-based) : K-means
* 밀도 기반(Density-based) : DBSCAN

계층적 군집분석 (Hierarchical Clustering)

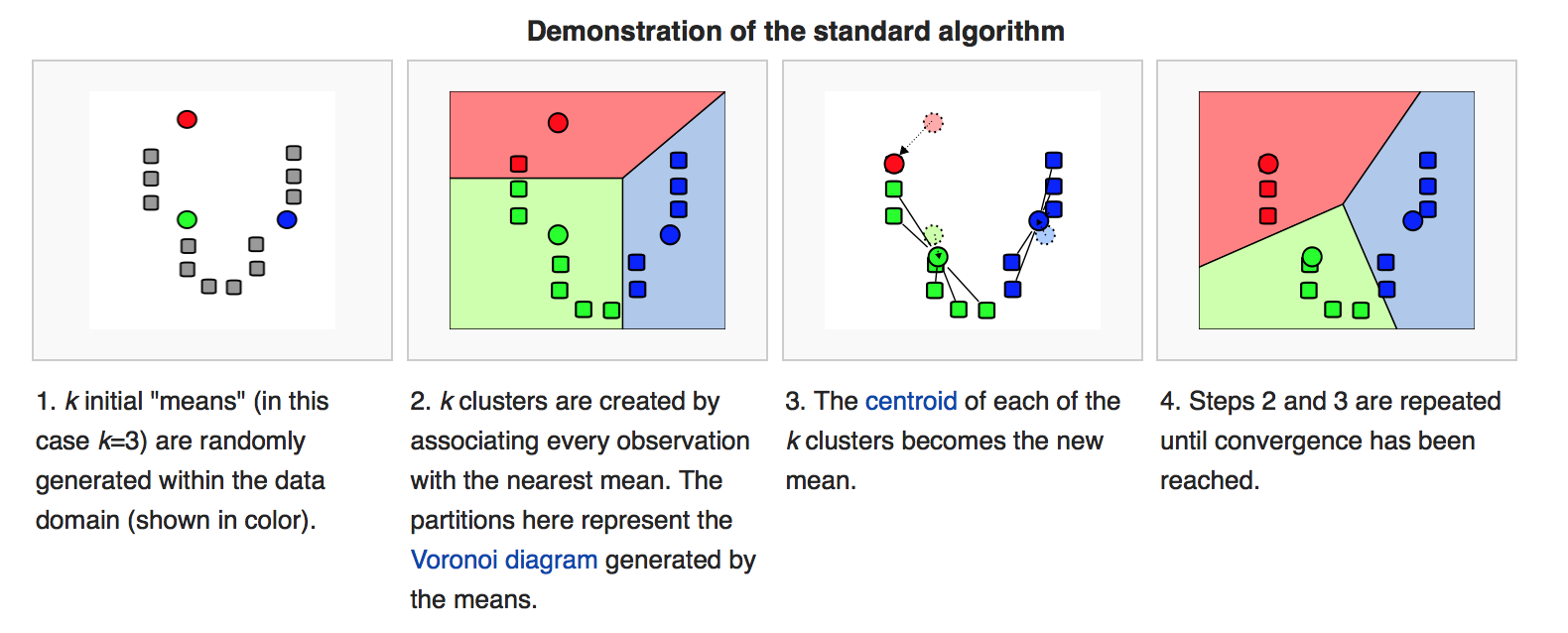
비계층적 군집 분석(Non-Hierarchical Clustering)

계층을 두지 않고 그룹화를 할 유사도 측정 방식에 따라 최적의 그룹을 계속적으로 찾아나가는 방법.

K-means

증신 기반 클러스터링 방법으로, “유사한 데이터는 중심점(centroid)을 기반으로 분포할 것이다”라는 가정을 기반으로 한다.

n개의 데이터와 k(<=n)개의 중심점(centroid)이 주어졌을 때 각 그룹 내의 데이터와 중심점 간의 비용(거리)을 최소화하는 방향으로 계속 업데이트를 해줌으로써 그룹화를 수행하는 기법.



1. 초기점(k) 설정

* k는 중심점. 묶일 그룹(cluster) 수.
* 예시에서 k=3 (동그라미 수)

1. 그룹(cluster) 부여

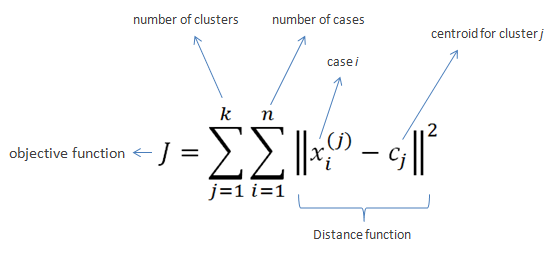
* k개의 중심점(●)과 개별 데이터(■) 간의 거리를 측정
* 가장 가까운 중심점으로 데이터 부여.

1. 중심점 업데이트

* 할당된 데이터들의 평균값으로 새로운 중심점을 업데이트한다.

1. 최적화

* 2, 3번의 작업을 반복적으로 수행.
* 변화 없으면 작업 중단



중심점 업데이트를 위해 평균값을 사용하기 때문에 K-means라고 불리며, 이상치에 영향을 받는 단점을 보와한기 위해 중간값을 활용한 K-medoids방법도 있다.

Python의 Scikit learn 라이브러리의 KMeans 함수로 구현 가능

KMeans(n\_cluster=, init=, random\_state=)

n\_cluster= k의 수

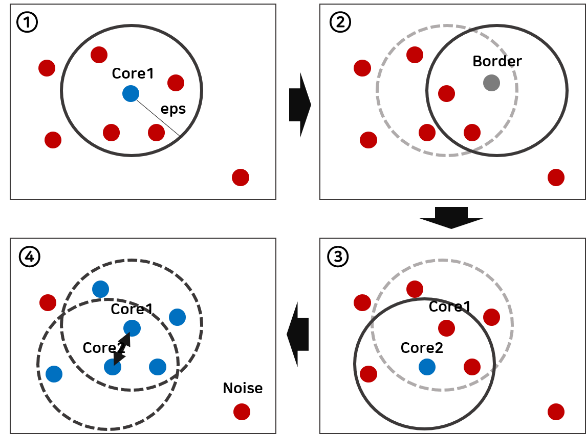
init= 초기값 지정(default = random)

random\_state= random 시드 설정

DBSCAN

밀도 기반 클러스터링 방법. “유사한 데이터는 서로 근접하게 분포할 것이다”는 가정을 기반.

K-means와 달리 그룹의 수(k)를 설정하지 않고 자동적으로 최적의 그룹 수를 찾아나간다.



1. 하나의 점( )을 중심으로 반경(eps) 내에 최소 점이 4개(minPts=4)이상 있으면, 하나의 군집으로 판단하여 해당 점( )은 core가 된다.
2. 반경 내에 점이 3개 뿐이므로 core가 되지 않지만 core1의 군집에 포함된 점으로, 이는 border가 된다.
3. 1번과 마찬가지로 core가 된다.
4. 반경내의 점 중에 core1이 포함되어 있어 군집으로 연결하여 하나의 군집으로 묶인다.

이와 같은 방식으로 군비의 확산을 반복하면서, 자동으로 최적의 군집수가 도출된다.

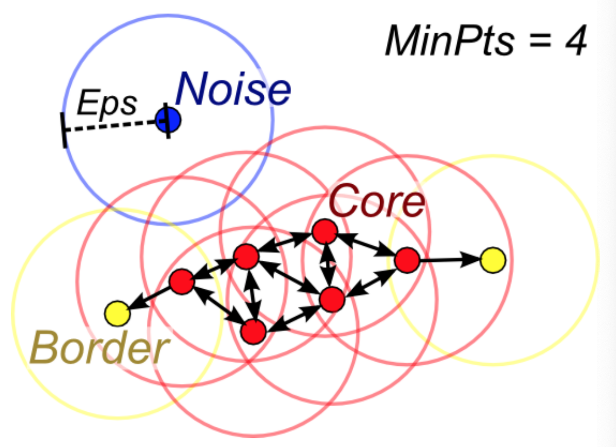
minPts: 반경 내 최소 개체 수

eps(epsilon): 군집화할 반경

Core: 중심점(minPts를 만족할 경우)

Border: 경계점(minPts를 만족하진 않지만, 어느 core 반경에 속한 경우)

Noise: 어느 군집에도 속하지 않는 점.



계층적 군집 분석

* 군집의 개수가 정해지지 않았을 때 사용. 몇 개의 군집으로 나누어야 하는지 결정하기 위해 사용하기도 함.
* 가까운 개체끼리 차례로 묶거나 멀리 떨어진 개체끼리 차례로 분리하는 방법으로, 합병에 의한 방법과 분할에 의한 방법이 있다.
* 구현이 간단하고 이해하기 쉬우며, 덴드로그램과 같은 그래프로 결과를 확인할 수 있다.
* 한번 병합된 개체는 다시 분리되지 않는다.
* 변수가 많거나 데이터의 크기가 클 경우, 계산량이 많아져 느리다.

합병에 의한 방법(Agglomerative methods)

* 가까운 개체끼리 묶어 군집을 이루어 나가는 것으로 마지막에는 하나의 군집이 되며 덴드로그램으로 표현하게 된다.

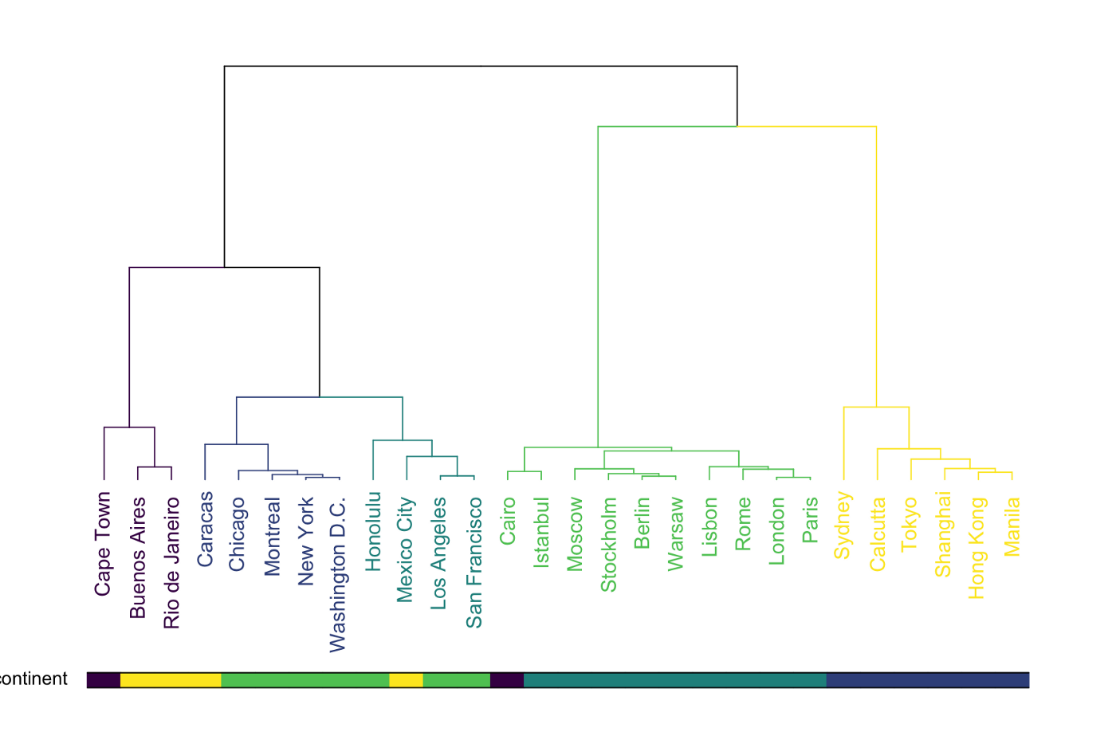
1. 1개의 개체를 가지는 샘플 사이즈 n개의 군집으로 시작한다.

2. 각각의 거리 매트릭스를 계산한다.

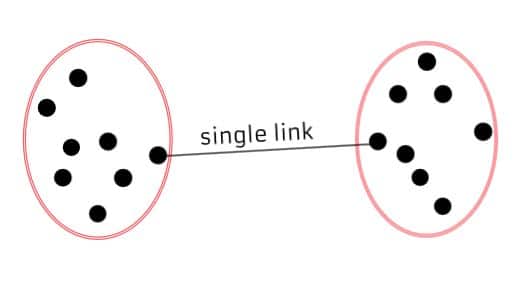
3. 가장 거리가 가까운 두 개의 군집을 찾아서 하나의 군집으로 묶는다.

4. 묶인 하나의 군집과 다른 개체의 거리를 업데이트한다.

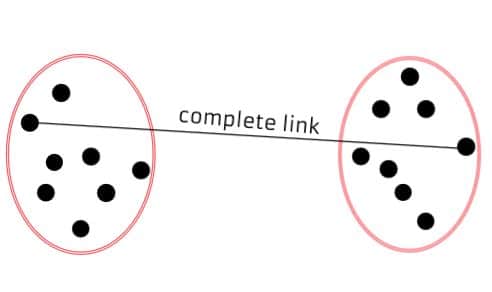
5. 모든 개체가 하나의 군집이 될 때까지 반복한다.



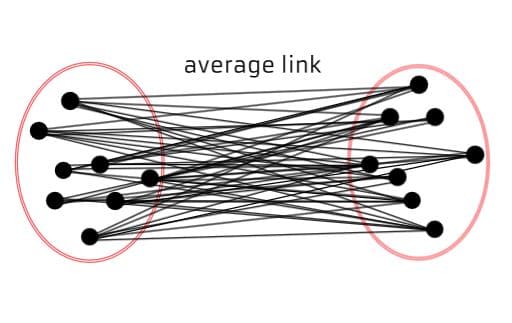
최단연결법(Single linkage method)



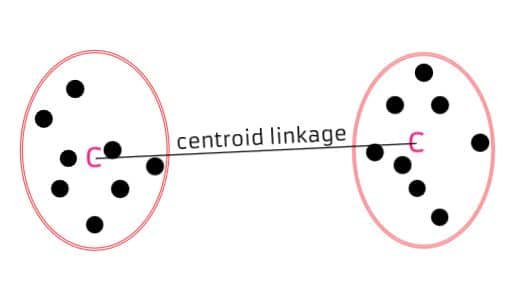
최장연결법(Complete linkage method)



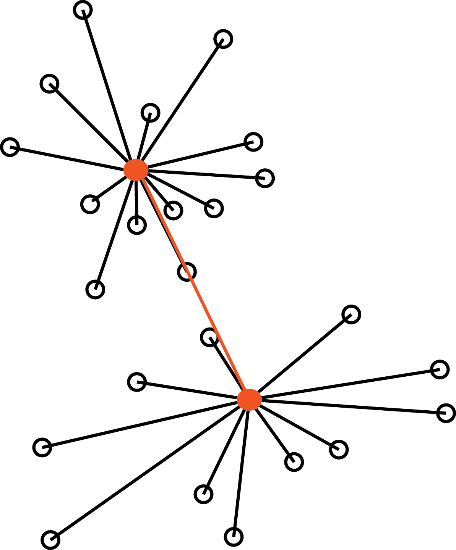
평균연결법(Average linkage method)



중심연결법(Centroid linkage method)



Wards연결법(Wards linkage method)



분할에 의한 방법(Devisive methods)

* 전체를 하나의 군집에서 두 개의 군집으로 분할하는 것에서 시작해서 마지막에는 각각의 개체들이 개별 군집으로 나누어지게 하는 방법.

<https://webnautes.tistory.com/803>