Week 2 스터디 발표

클러스터링(Clustering) 기법들 소개

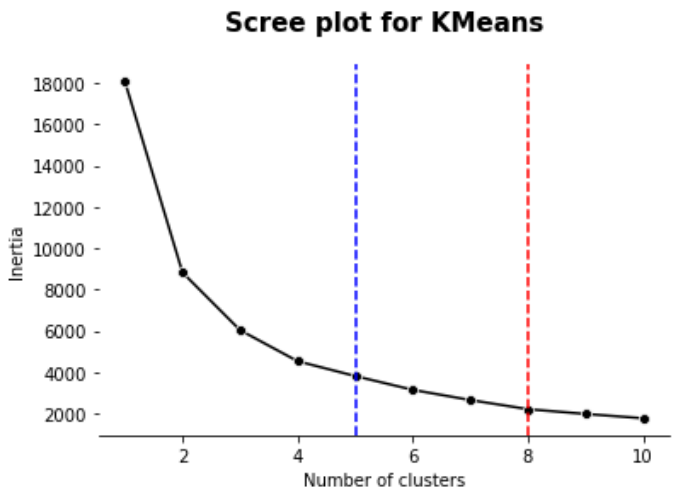
Kmeans Clustering

알고리즘

1) 각 feature를 표준화
2) Train data에서 임의로 K개의 표본을 뽑아 K개의 클러스터의 centroid를 설정한다
3) 각 데이터의 유클리디안 거리를 기준으로 가장 가까운 centroid에 해당하는 클러스터에 편입한다
4) 각 클러스터의 평균을 새로 구해 이를 새로운 centroid로 설정하고 3)을 진행한다
5) 각 클러스터의 member가 변하지 않을 때까지 혹은 미리 정해진 최대 반복 횟수에 도달할 때까지 4)를 반복한다

단점 1) Kmeans 알고리즘의 최대 단점은 클러스터의 개수 K를 임의로 설정한다는 것

→ **Elbow Method**를 사용하여 최적의 K를 결정한다.



[Scree plot]

- inerita: Kmeans로 데이터가 얼마나 잘 클러스터링 됐는지를 나타내는 척도로, 각 클래스에서 데이터와 centroid의 거리의 제곱합이다.
- inertia의 감소속도가 확연히 줄어드는 K를 찾아 클러스터의 개수로 설정한다.

Kmeans++ Clustering

단점 2) Kmeans에서는 초기 centroid를 랜덤하게 부여하는데, 이 초기 centroid에 따라 클러스터링 결과가 달라질 수 있다는 문제점이 있다.

→ Kmeans++ 알고리즘 사용하여 초기 centroid를 지정하는 방법을 수정한다.

```
inertia = []
k_list = range(1, 11)
for k in k_list:
    cluster = KMeans(n_clusters = k)
    cluster.fit(coords)
    inertia.append(cluster.inertia_)
```

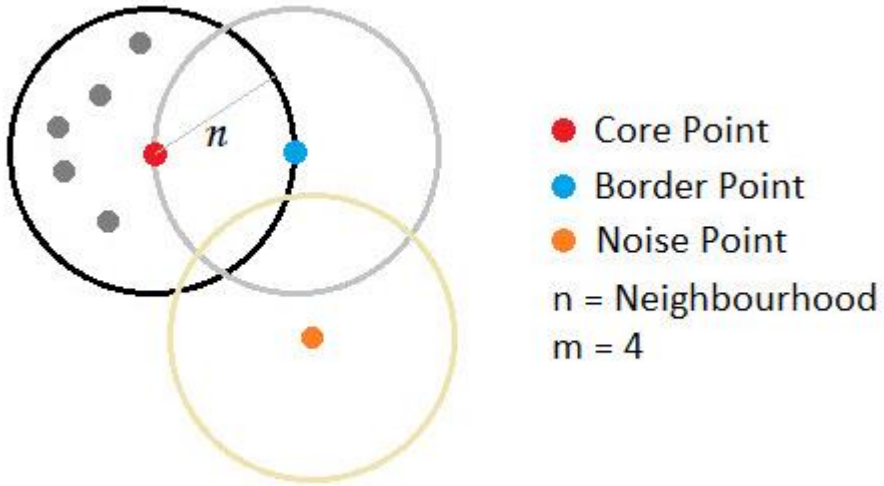
옵션 지정
init – 초기 centroid 설정 방법 → k-means++, random

DBSCAN

밀도 기반 클러스터링

공간상에 높은 밀도를 가지고 모여 있는 관측치들을 하나의 그룹으로 간주하고, 낮은 밀도를 가지고 있는 관측치는 이상치 혹은 잡음(noise)로 분류한다.

관측치 유형 분류



Core point (핵심자료)
$\varepsilon - neighborhood$ 에 M개 이상의 다른 관측치를 포함하는 관측치

Border point (주변자료)
핵심자료는 아니지만 $\varepsilon - neighborhood$ 에 핵심자료를 포함하는 관측치

Noise point (잡음자료)
핵심자료도 주변자료도 아닌 관측치

Parameter
- ε (eps): 너무 작으면 많은 관측치가 noise로 분류되고, 너무 크면 클러스터의 개수가 적어진다.
- M(min_samples): 클러스터의 최소 크기를 결정하는 모수