군집화(Clustering)

: 데이터 포인트들을 별개의 군집으로 그룹화하는 것.

유사성이 높은 데이터들을 동일한 그룹으로 분류하고, 서로 다른 군집들이 상이성을 가지도록 그룹화한다.

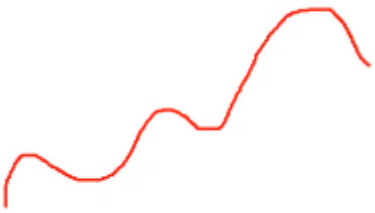
군집화 활용 분야

- 세분화(고객, 마켓, 브랜드, 사회 경제 활동)
- 세분화(Image 검출, Tracking)
- 이상 검출(Abnomaly Detection)

어떤 요소들을 유사성으로 정의?

군집화 알고리즘 종류

- K-Means (Centroid 기반)
- Mean Shift (Centroid 기반)
- GMM(Gaussian Mixture Model. 어느 정규 분포에 속하는지를 기반)

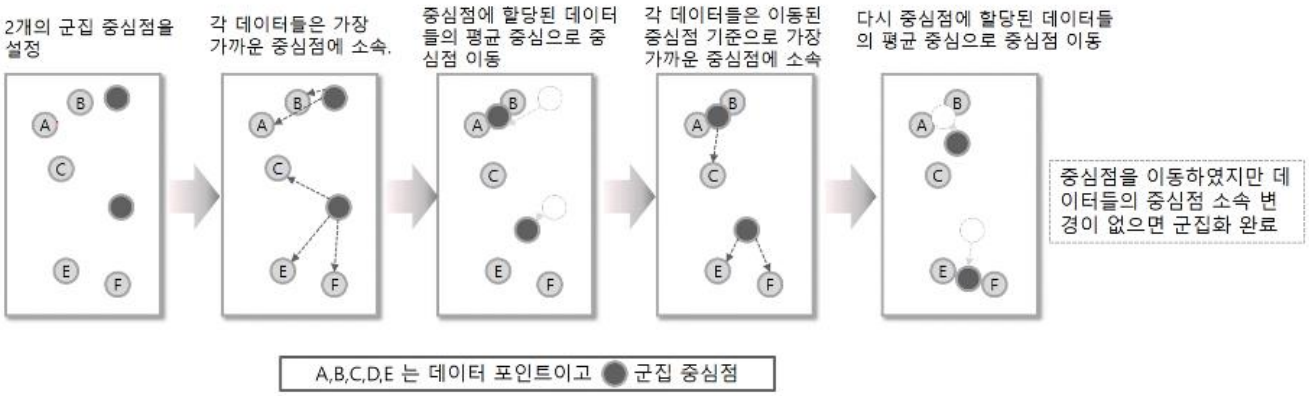


- DBSCAN (밀도에 기반)

K-Means

의문점 : 중심점은 어떻게 설정하는가?

군집 중심점(Centroid) 기반 클러스터링



1) 2개의 중심점 설정.
2) 각 데이터들은 가장 가까운 중심점에 소속.
3) 유클리디안 거리를 계산해 중심점에 할당된 데이터들의 평균 중심으로 중심점 이동
4) **중심점을 이동하였으나 소속 변경이 없을 때까지 2), 3)을 반복함**
5) 군집화 완료

K-means의 장단점

장점 :

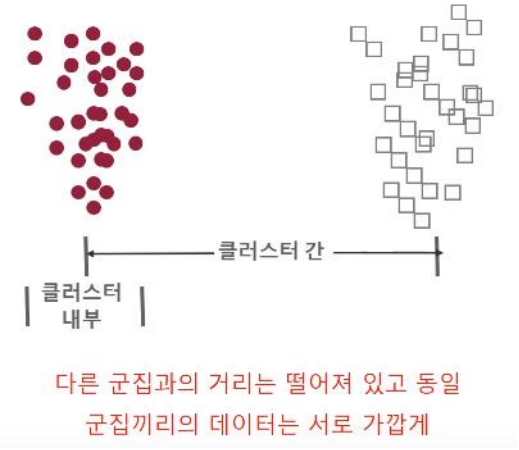
- 일반적인 군집화에서 가장 많이 사용되는 알고리즘
- 쉽고 간결
- **대용량 데이터에도 활용 가능**

단점 :

- **거리 기반** 알고리즘(Euclidean Distance)으로 속성의 개수가 매우 많을 경우 군집화 정확도가 떨어진다.(이를 위해 PCA로 차원 축소 적용 필요할 수 있음)
- 반복을 수행하는데 반복 횟수가 많을 경우 수행 시간이 느려짐
- 이상치(Outlier) 데이터에 취약함(Centroid가 이상한 곳에 갈 수 있음.)

군집 평가 – 실루엣 분석

: Label이 따로 주어지지 않은 경우 시각적으로 이를 판단해야 한다.



다른 군집과의 거리는 떨어져 있고 동일
군집끼리의 데이터는 서로 가깝게

- 실루엣 분석은 각 군집 간의 거리가 얼마나 효율적으로 분리되어 있는지를 나타낸다.
- 실루엣 분석은 개별 데이터가 가지는 군집화 지표인 **실루엣 계수**(Silhouette Coefficient)를 기반으로 한다.
- **개별 데이터가 가지는** 실루엣 계수는 해당 데이터가 같은 군집 내의 데이터와 얼마나 가깝게 군집화되어 있고, 다른 군집에 있는 데이터와는 얼마나 멀리 분리되어 있는지를 나타내는 지표이다.

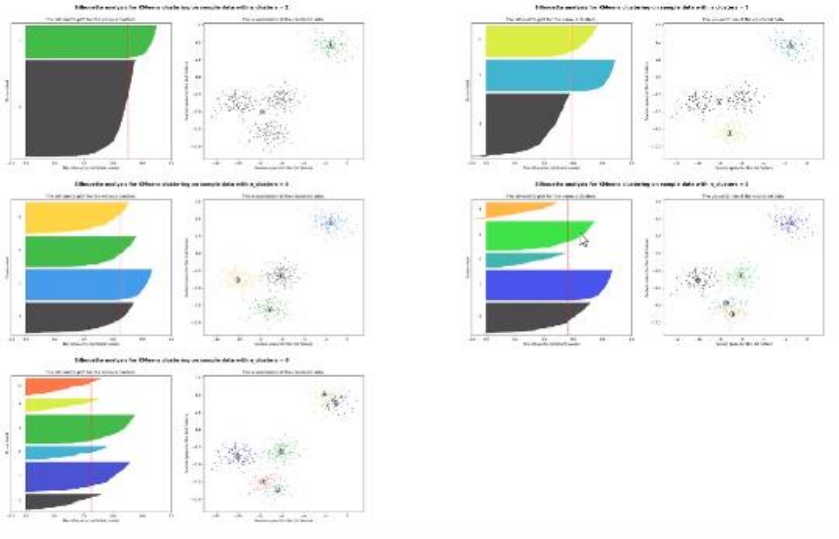
실루엣 계수

$$s(i) = \frac{(b(i) - a(i))}{(\max(a(i), b(i))}$$

- aij는 i번째 데이터에서 자신이 속한 클러스터 내의 다른 데이터 포인트(j번째 데이터)까지의 거리
- ai는 i번째 데이터에서 자신이 속한 클러스터내의 다른 데이터 포인트들과의 거리 평균.
- bi도 마찬가지.
- 두 군집간의 거리가 얼마나 떨어져 있는가는 b(i) – a(i)이며, 이 값을 정규화하기 위해 MAX(a(i), b(i)) 값으로 나눈다.
- 실루엣 계수는 **-1에서 1까지의 값**을 가지며
  - 1로 가까워질수록 근처의 군집과 더 멀리 떨어져 있다는 것
  - 0에 가까울수록 근처의 군집과 가까워 진다는 것
  - - 값은 다른 군집에 데이터 포인트가 할당되어 있다는 것을 나타냄.

실루엣 분석을 적용할 수 있는 군집 기준

- 전체 실루엣 계수의 평균 값이 0 ~ 1사이 값을 가지며, 1에 가까울수록 좋다.
- 전체 실루엣 계수의 평균 값과 더불어 개별 군집의 평균값의 편차가 크지 않아야 한다.
- 대용량 데이터에는 적합하지 않음. 수행 시간이 매우 오래 걸린다.

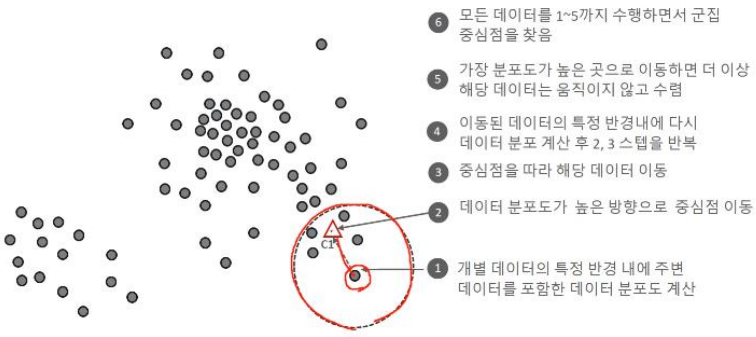


Cluster 개수를 늘이면서 실루엣 계수 값에 따른 분포를 볼 수 있다.(출처 : 사이킷런 홈페이지)

```
For n_clusters = 2 The average silhouette_score is : 0.7049787496083262
For n_clusters = 3 The average silhouette_score is : 0.5882004012129721
For n_clusters = 4 The average silhouette_score is : 0.6505186632729437
For n_clusters = 5 The average silhouette_score is : 0.5745566973301872
For n_clusters = 6 The average silhouette_score is : 0.4387644975296138
```

실루엣 계수가 높다고 해도 평균에 대한 값이기 때문에 Cluster 내에 속한 특정 개별 요소에 대해서는 굉장히 불명확하게 Clustering 되어 있을 수도 있다.

Mean Shift



1) 개별 데이터의 특정 반경 내에 주변 데이터를 포함한 데이터 분포도 계산

2) **데이터 분포도가 높은 방향으로 중심점 이동**

3) **중심점을 따라 해당 데이터 이동**

4) 이동된 데이터의 특정 반경 내에 다시 데이터 분포 계산 후 **2), 3) 스텝을 반복**

5) 가장 분포도가 높은 곳으로 이동하면 더 이상 해당 데이터는 움직이지 않고 수렴

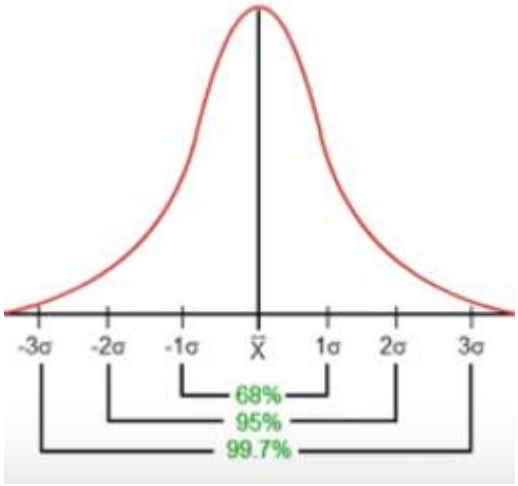
6) 모든 데이터를 1~5까지 수행하면서 군집 중심점을 찾음

하나 하나씩 수행

특정 데이터가 반경 내의 데이터 분포 확률 밀도가 가장 높은 곳으로 이동할 때, 주변 데이터들과의 거리 값을 Kernel 함수 값으로 입력한 뒤, 그 반환 값을 현재 위치에서 Update하면서 이동

KDE(Kernel Density Estimation)

(Kernel 함수 : Y축을 중심으로 대칭이면서 적분 값이 1인 Non-negative 함수)



: KDE는 커널(Kernel)함수를 통해 어떤 변수의 확률 밀도 함수를 추정하는 방식. 관측된 데이터 각각에 커널 함수를 적용한 값을 모두 더한 뒤 데이터 건수로 나누어서 확률 밀도 함수를 추정.

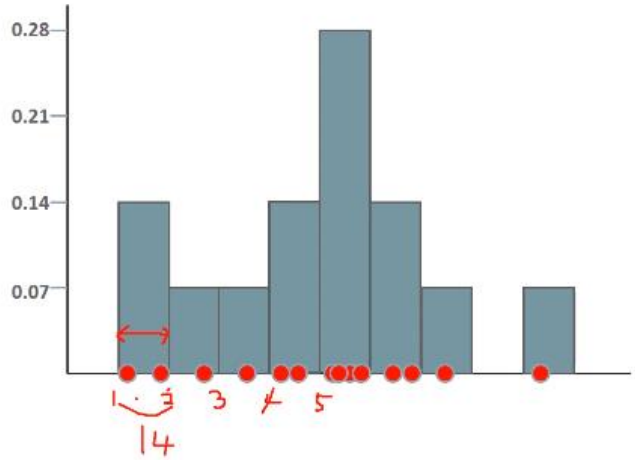
- 확률 밀도 함수(PDF : Probability Density Function) : 확률 변수의 분포를 나타내는 함수. 대표적으로 정규 분포, **감마 분포, t-분포** 등이 있다.
- 확률 밀도 함수를 알게 되면 특정 변수가 어떤 값을 갖게 될지의 확률을 알게 됨. 이는 확률 밀도 함수를 통해 변수의 특성(예를 들어 정규 분포의 평균, 분산, 확률 분포 등)에 해당하는 많은 요소를 알 수 있게 됨.

확률 밀도 추정 방법

- 모수적 추정(Parametric) : 데이터가 특정 데이터 분포(예를 들어 가우시안 분포)를 따른다는 가정 하에 데이터 분포를 찾는 방법. **Gaussian Mixture** 등이 있다.
- 비모수적 추정(Non-Parametric) : 데이터가 특정 분포를 따르지 않는다는 가정 하에서 밀도를 추

정. 관측된 데이터만으로 확률 밀도를 찾는 방법으로써 대표적으로 KDE가 있다.

비모수적 밀도 추정 - Histogram(히스토그램)



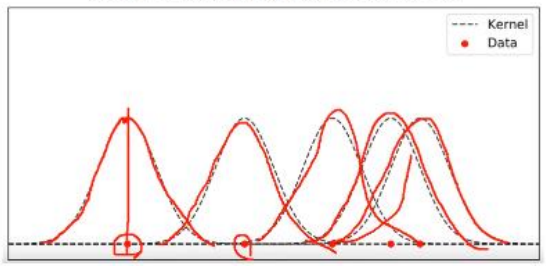
히스토그램 밀도 추정의 문제점.

- Bin의 경계에서 불연속성이 나타남
- Bin의 크기에 따라 히스토그램이 달라짐.

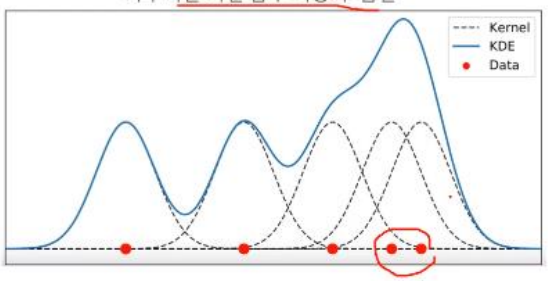
비모수적 밀도 추정 - KDE

: KDE는 개별 관측 데이터들에 커널 함수를 적용한 뒤, 커널 함수들의 적용 값을 모두 합한 뒤에 개별 관측 데이터의 건수로 나누어서 확률 밀도 함수를 추정하는 방식. 커널 함수로는 대표적으로 가우시안 분포 함수가 사용됨.

개별 관측 데이터에 가우시안 커널 함수 적용



가우시안 커널 함수 적용 후 합산



KDE는 아래와 같은 커널함수 식으로 표현됨. 이때 K는 커널함수, x는 random variable, xi는 관측값, h는 bandwidth

$$KDE = \frac{1}{n}\sum_{i=1}^{n} K_h(x-x_i) = \frac{1}{nh}\sum_{i=1}^{n} K\left(\frac{x-x_i}{h}\right)$$

대표적인 커널함수는 가우시안 분포임. $f(x \mid \mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$

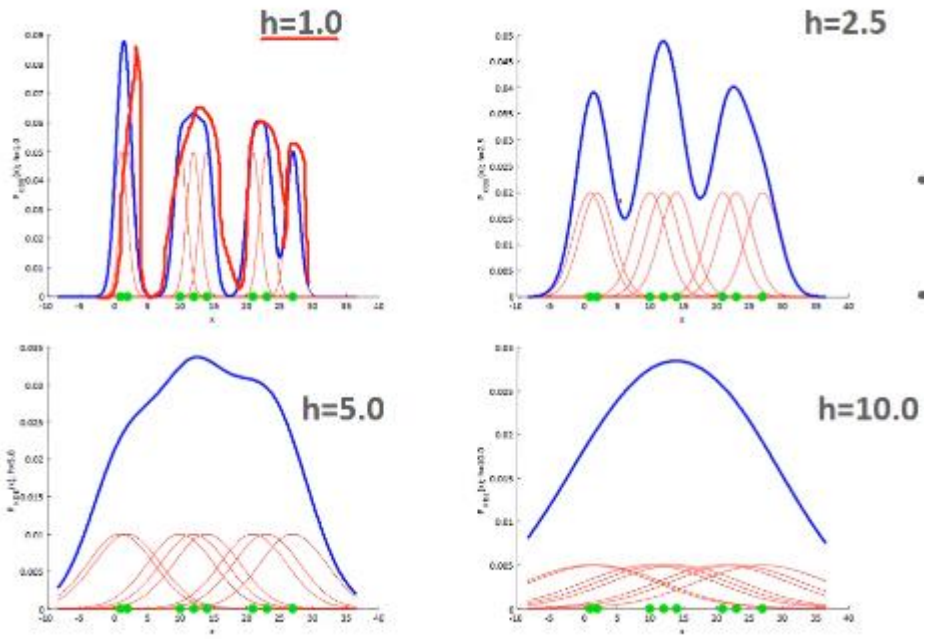
가우시안 커널함수를 적용한 KDE는 아래와 같음. 이 경우 관측값 xi는 평균, bandwidth h는 표준편차와 동일.

$$KDE = \frac{1}{nh}\sum_{i=1}^{n} \frac{1}{\sqrt{2\pi}h} e^{\left(-\frac{1}{2}\left(\frac{x-x_i}{h}\right)^2\right)}$$

가우시안 커널함수를 적용할 경우 최적의 bandwidth는 아래와 같습니다.

$h = \left(\frac{4\sigma^5}{3n}\right)^{\frac{1}{5}} \approx 1.06\sigma n^{-1/5}$ 단, n은 샘플 데이터의 개수, σ는 샘플 데이터의 표준편차

h인 bandwidth가 작으면 표준편차가 작은 것이므로 그래프가 완만하게, bandwidth가 크면 그래프가 깎아지게 형성됨.

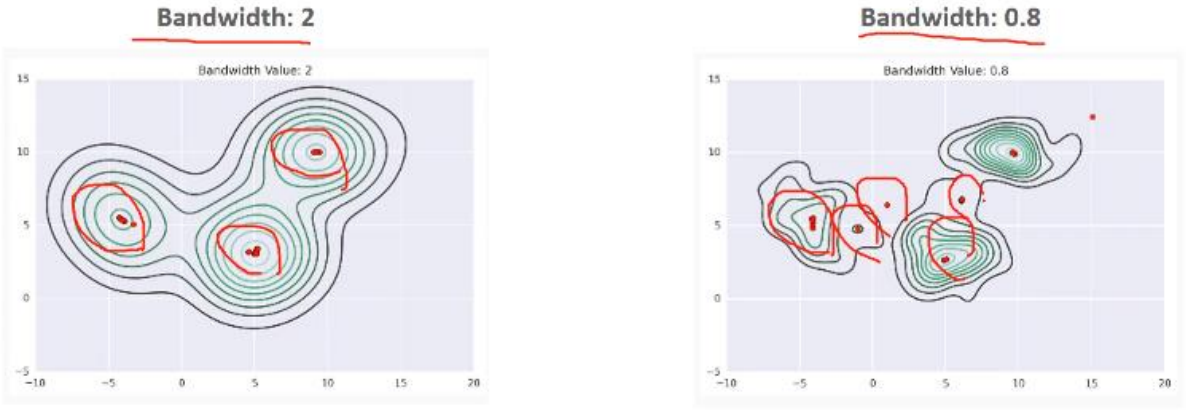


h가 작으면 overfitting 되기 쉬우며, h가 크면 과도하게 smoothing(평균치를 이용하여 보정하는 평활화 작업) 되어 underfitting 되기 쉽다.

Bandwidth에 따른 KDE의 변화

Mean Shift는 Bandwidth가 클수록 적은 수의 클러스터링 중심점을, Bandwidth가 작을수록 많은 수의 클러스터링 중심점을 갖게 된다.(위의 그림과 기본적인 개념을 이용하여 이해)

또한, Mean Shift는 군집의 개수를 지정하지 않으며, 오직 Bandwidth의 크기에 따라 군집화를 수행



Mean Shift에서 **가장 중요한 초기화 파라미터는 Bandwidth**이다. 이를 어떻게 설정하느냐에 따라 군집화 성능이 크게 달라지게 된다.

최적의 bandwidth 계산을 위해 사이킷런은 estimate_bandwidth() 함수를 제공한다.

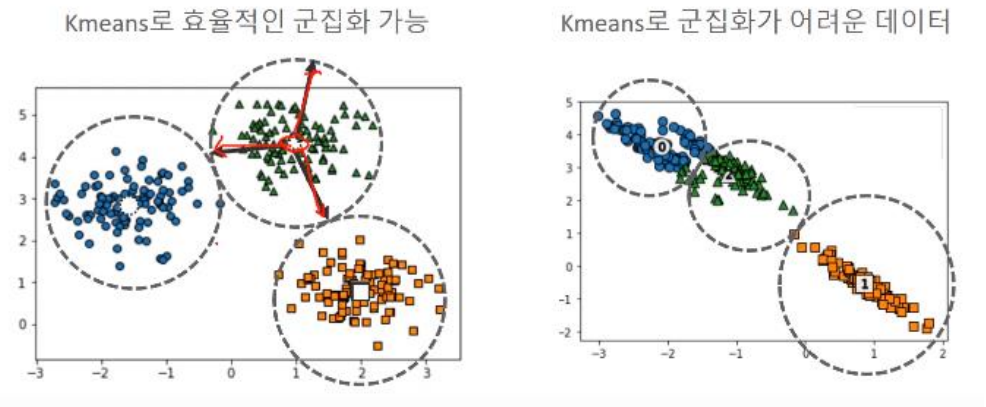
KDE를 시각화하여 나타내는 것에는 seaborn 이 유용하다.

단점 :

- bandwidth에 매우 민감한 단점이 있다.

datamining보다 영상, object tracking 하는데 많이 사용된다.

GMM (Gaussian Mixture Model) – 거리 기반 K-Means의 문제점.



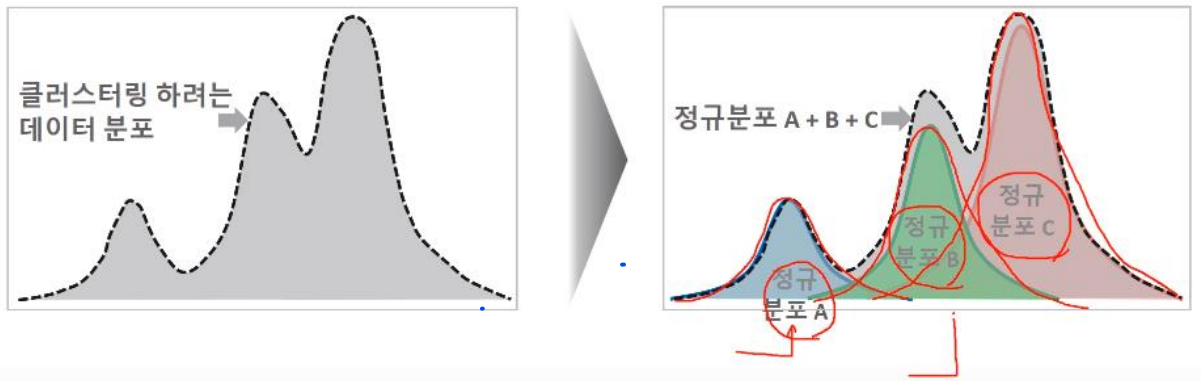
K-means는 특정 중심점을 기반으로 거리적으로 퍼져 있는 데이터 세트에 군집화를 적용하면 효율적이다.

하지만, K-means는 이러한 데이터 분포를 가지지 않는 데이터 세트에 대해서는 효율적인 군집화가 어렵다.

GMM 군집화

- GMM 군집화는 군집화를 적용하고자 하는 데이터가 **여러 개의 다른 가우시안 분포(Gaussian Distribution)을 가지는 모델로 가정**하고 군집화를 수행한다.

  가령 **1000개의 데이터 세트가 있다면 이를 구성하는 여러 개의 정규 분포 곡선을 추출**하고, **개별 데이터가 이 중 어떤 정규분포에 속하는지 결정**하는 방식이다.



어느 정규 분포에 속하는지 찾기 위해서는 GMM의 모수 추정이 필요.

GMM 모수 추정 : 개별 정규 분포들의 **평균**과 **분산**, 그리고 **데이터가 특정 정규 분포에 해당될 확률을 추**

정하는 것.

GMM 모수 추정을 위한 **EM**(Expection and Maximization)

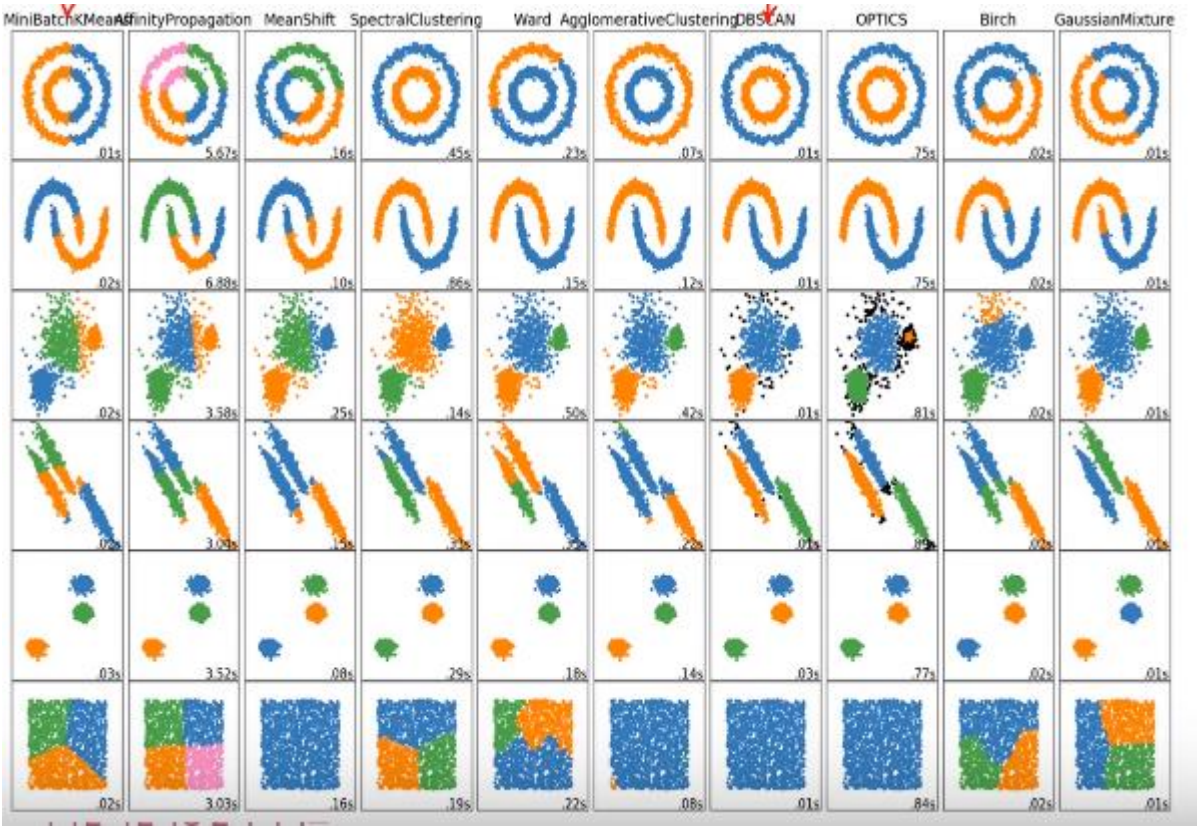
- Expectation : 개별 데이터 각각에 대해서 특정 정규 분포에 소속될 **확률**을 구하고, 가장 높은 확률을 가진 정규 분포에 소속
- Maximization : 데이터들이 특정 정규분포로 소속되면 **다시 해당 정규분포의 평균과 분산을 구함.** 해당 데이터가 발견될 수 있는 가능도를 최대화(Maximum Likelihood)할 수 있도록 평균과 분산(모수)를 구함.

EM 반복을 수행하다가 개별 정규분포의 **모수인 평균과 분산이** 더 이상 변경되지 않고 **각 개별 데이터들이 이전 정규 분포 소속에서 더 이상 변경되지 않으면** 그것으로 최종 군집화를 결정.

주요 파라미터는 Mixture Model의 개수. 즉, 군집화 개수

DBSCAN

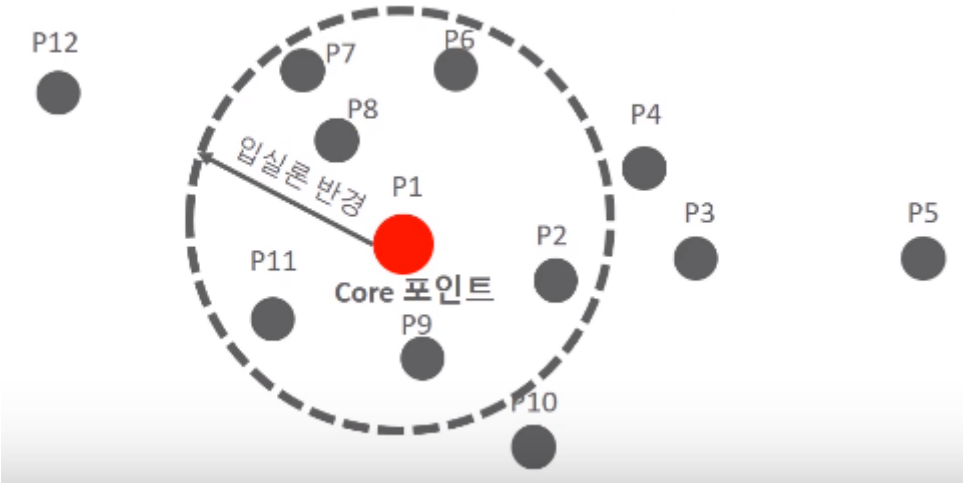
: 특정 공간 내에 데이터 밀도 차이를 기반으로 군집을 생성. 사용자가 군집 개수 지정 불가,



단점 :

- 데이터의 밀도가 자주 변하거나 아예 모든 데이터의 밀도가 크게 변하지 않으면 군집화 성능이 떨어진다.

- 피처의 개수가 많으면 군집화 성능이 떨어진다.



DBSCAN을 구성하는 가장 중요한 두 가지 파라미터는 입실론**(epsilon)**으로 표기하는 주변 영역과 이 입실론 반경 내에 최소 데이터의 개수**(min point)**이다. min point에는 자기 자신은 포함하지 않음.

입실론에 해당하는 각 포인트마다 이름이 있음

- 핵심 포인트(Core Point) : 주변 영역 내에 **최소 데이터 개수 이상의 타 데이터**를 가지고 있을 경우 해당 테이터를 핵심 포인트라고 함.
- 이웃 포인트(Neighbor Point) : 주변 영역 내에 위치한 타 데이터
- 경계 포인트(Border Point) : 주변 영역 내에 최소 데이터 개수 이상의 이웃 포인트를 가지고 있지 **않지만 핵심 포인트를 이웃 포인트**로 가지고 있는 데이터
- 잡음 포인트(Noise Point) : 최소 데이터 개수 이상의 이웃 포인트를 가지고 있지 않으며, 핵심 포인트도 이웃 포인트로 가지고 있지 않은 데이터를 잡음 포인트라고 한다.

핵심포인트를 연결하면서 군집화(Clustering) 구성.

eps : 입실론 주변 영역의 반경, min_samples : min point와 자기 자신

출처 : 인프런 머신러닝 완벽가이드