클러스터 분석: 기본 개념과 방법론

최우진

2019/07/31

10.3 Hierarchical Methods

- 계층적 클러스터링은 데이터 오브젝트를 구조, 즉 ‘트리’ 형태의 클러스터로 나눈다.
- ‘트리’ 형태의 클러스터는 정리와 시각화에 편리하다.
- 종류: 조적식(agglomerative), 분할식(divisive)
- 클러스터링 품질을 높이는 방법은 다른 클러스터링 기법과 조합해서 ‘다중 단계 클러스터링’이다.
- 다중 단계 클러스터링의 종류: BIRCH기법, Chameleon 기법

10.3.3 BIRCH: Multiphase Hierarchical Clustering Using Clustering Feature Trees

- BIRCH(Balanced Iterative Reducing and Clustering using Hierarchies)은 대규모 정량 데이터의 클러스터링을 목적으로 만들어짐.
- '규모 확장' 과 '중간 과정을 돌이킬 수 없는 문제' 해결.
- 클러스터링 특성 (Clustering feature) 이란 개념을 통해 클러스터를 종합.
- CF-트리 (Clustering feature tree)로 클러스터의 구조를 보여줌.
- 식

$$CF = \langle n, LS, SS \rangle,$$

$$LS : \sum_{i=1}^{N} \vec{X}_i \qquad SS : \sum_{i=1}^{N} \vec{X}_i^2$$

클러스터링 특성: 이 공간상의 오브젝트 클러스터 세트 정보를 종합한 3D벡터
N: 데이터 점 개수

10.3.3 BIRCH: Multiphase Hierarchical Clustering Using Clustering Feature Trees

- 클러스터링 특성
- 중앙자(x_0), 반경(R), 직경(D)

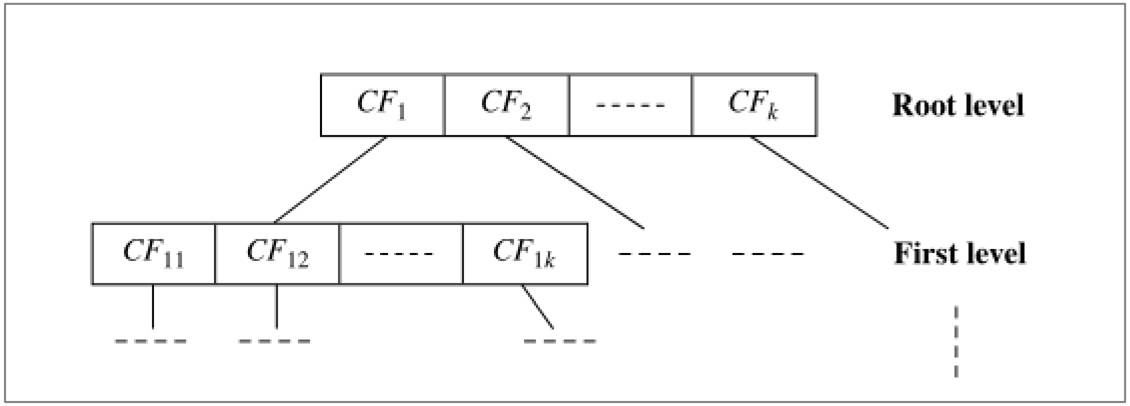
$$\mathbf{x_0} = \frac{\sum_{i=1}^{n} \mathbf{x_i}}{n} = \frac{LS}{n},$$

$$R = \sqrt{\frac{\sum_{i=1}^{n} (\mathbf{x_i} - \mathbf{x_0})^2}{n}} = \sqrt{\frac{nSS - 2LS^2 + nLS}{n^2}},$$

$$D = \sqrt{\frac{\sum_{i=1}^{n}\sum_{j=1}^{n} (\mathbf{x_i} - \mathbf{x_j})^2}{n(n-1)}} = \sqrt{\frac{2nSS - 2LS^2}{n(n-1)}}.$$

10.3.3 BIRCH: Multiphase Hierarchical Clustering Using Clustering Feature Trees

- CF-트리 구조



$$CF_1 = \langle n_1, LS_1, SS_1 \rangle \text{ and } CF_2 = \langle n_2, LS_2, SS_2 \rangle,$$

$$CF_1 + CF_2 = \langle n_1 + n_2, LS_1 + LS_2, SS_1 + SS_2 \rangle.$$

CF-tree에 적용하는 파라미터
- **분기 기준**(Branching Factor) **B**
- **역치**(Threshold) **T**

10.3.3 BIRCH: Multiphase Hierarchical Clustering Using Clustering Feature Trees

- 알고리즘

  1. 모든 Data를 읽어서 초기 메모리에서 CF tree 생성.

  2. 더 작은 CF tree로 만들어 바람직한 길이로 압축.

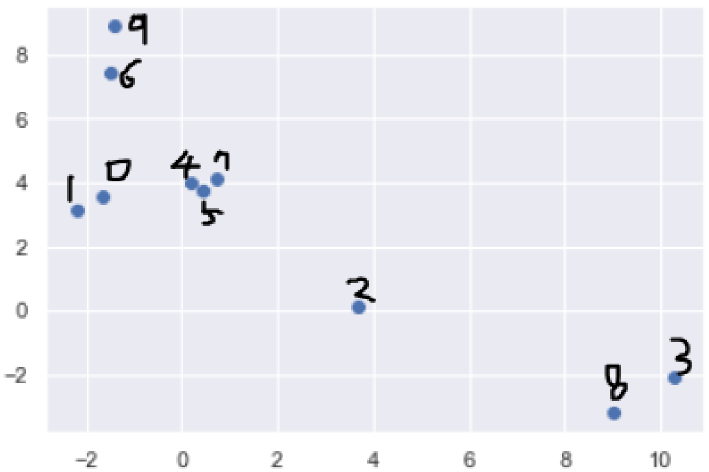
  3. 글로벌 클러스터링 진행.

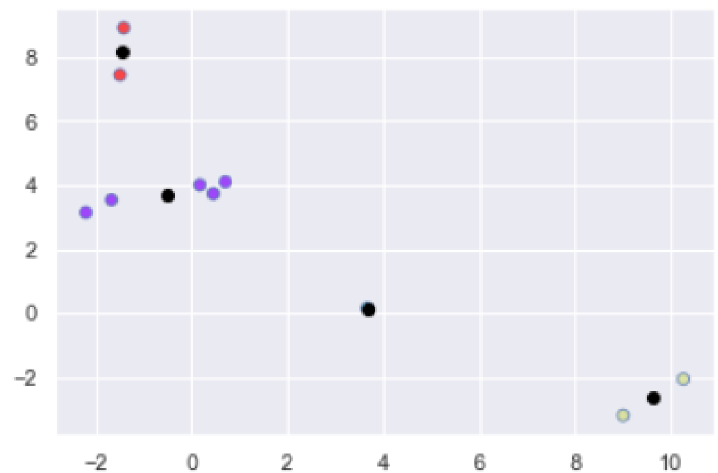
  4. 클러스터링 정제.

10.3.3 BIRCH: Multiphase Hierarchical Clustering Using Clustering Feature Trees

- 예시

```
[[-1.67173659  3.5340075 ]
 [-2.20667721  3.1382543 ]
 [ 3.66800921  0.15565258]
 [10.26934366 -2.05390901]
 [ 0.17304202  3.9973133 ]
 [ 0.4549417   3.72528035]
 [-1.49147123  7.42857208]
 [ 0.70514131  4.0921754 ]
 [ 9.01065223 -3.18950696]
 [-1.41766966  8.89824283]]
```



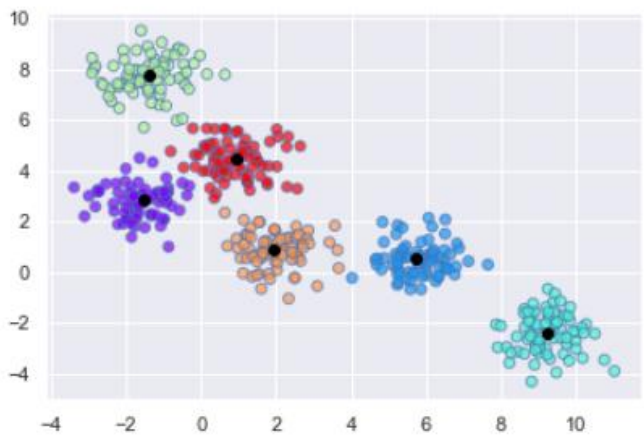


10.3.3 BIRCH: Multiphase Hierarchical Clustering Using Clustering Feature Trees

- 예시

```
%matplotlib inline
X, clusters = make_blobs(n_samples=450, centers=6, cluster_std=0.7, random_state=0)
plt.scatter(X[:,0], X[:,1], alpha=0.7, edgecolors='b')
```

```
<matplotlib.collections.PathCollection at 0x2552c3fc7f0>
```




```
T = 1.5
B = 50
brc = Birch(branching_factor=B, n_clusters=None, threshold=T)
brc.fit(X)
```

```
Birch(branching_factor=50, compute_labels=True, copy=True, n_clusters=None,
   threshold=1.5)
```

```
labels = brc.predict(X)
print(max(labels))
```

```
5
```

10.3.4 Chameleon: Multiphase Hierarchical Clustering Using Dynamic Modeling

- 카멜레온(Chameleon)은 동적 모델링을 통해 두 클러스터 사이의 유사성을 측정하는 구조적 클러스터링 알고리즘
- 유사성 측정 방법
    1. 클러스터 내부의 오브젝트가 얼마나 잘 연결되었는지
    2. 클러스터들이 서로 얼마나 가까이 있는지
- 즉, 두개의 클러스터의 상호연결성이 높고 서로 가까이 있으면 하나의 클러스터로 결합

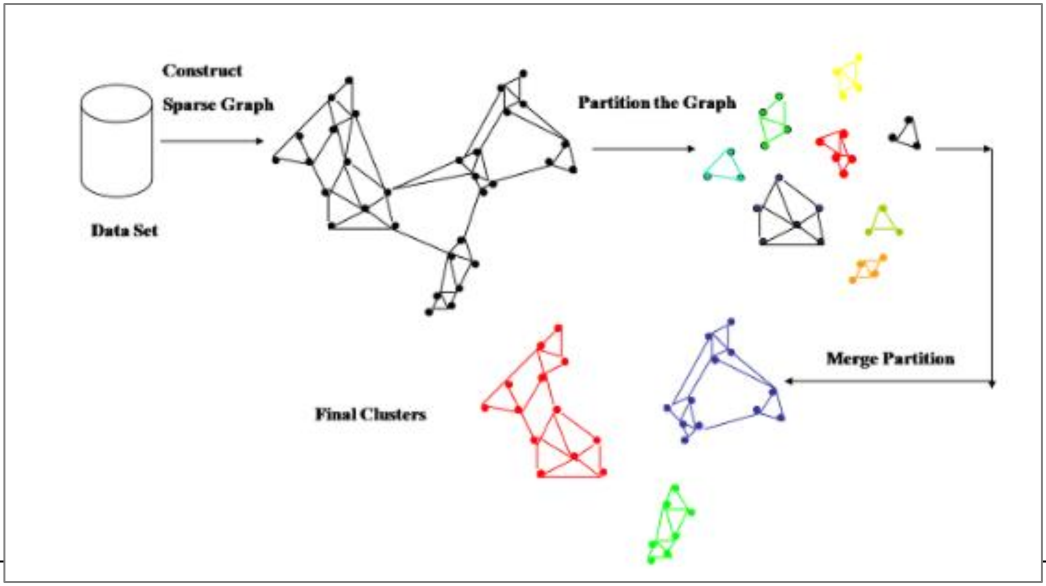
10.3.4 Chameleon: Multiphase Hierarchical Clustering Using Dynamic Modeling

- 알고리즘

    1. k-인접 이웃 그래프 기법으로 그래프 구조를 구성.

    2. 그래프 분할 알고리즘을 통해 연결 단절을 최소화하는 작은 서브 집합으로 분할.

    3. 반복적으로 하위 집합을 결합하여 잘 맞는 클러스터 탐색.



10.3.4 Chameleon: Multiphase Hierarchical Clustering Using Dynamic Modeling

- 반복적으로 하위 집합을 결합하여 잘 맞는 클러스터 탐색.
- 상호연결성 상대값(RI)

$$RI(C_i, C_j) = \frac{|EC_{\{C_i, C_j\}}|}{\frac{1}{2}(|EC_{C_i}| + |EC_{C_j}|)},$$

- 인접성 상대값(RC)

$$RC(C_i, C_j) = \frac{\overline{S}_{EC_{\{C_i, C_j\}}}}{\frac{|C_i|}{|C_i|+|C_j|}\overline{S}_{EC_{C_i}} + \frac{|C_j|}{|C_i|+|C_j|}\overline{S}_{EC_{C_j}}},$$

10.3.5 Probabilistic Hierarchical Clustering

- 알고리즘식 구조 클러스터링 단점

  1. 좋은 거리 측정 기준 모호.

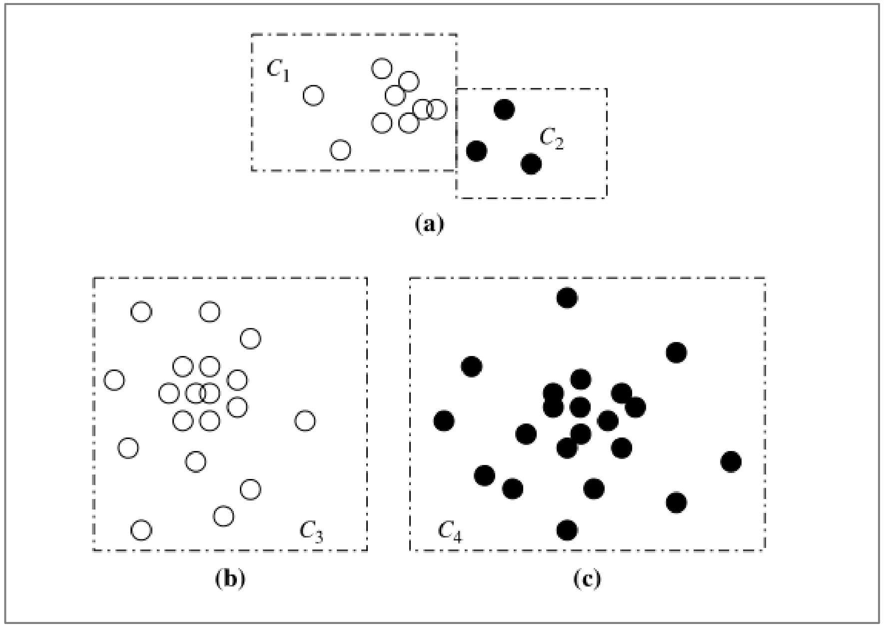
  2. 결손이 된 데이터 오브젝트의 속성 값 영향이 큼.

  3. 휴리스틱 방법이므로 클러스터링 구조 최적화 목적에 부합하지 않음.

- 이런 단점을 해소하고 클러스터 사이의 거리 측정에 확률 모델을 사용한 방법이 '확률식 구조 클러스터링'이다.

10.3.5 Probabilistic Hierarchical Clustering

- 예시

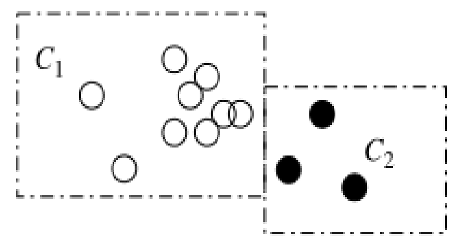


$$Q(\{C_1, \ldots, C_m\}) = \prod_{i=1}^{m} P(C_i),$$

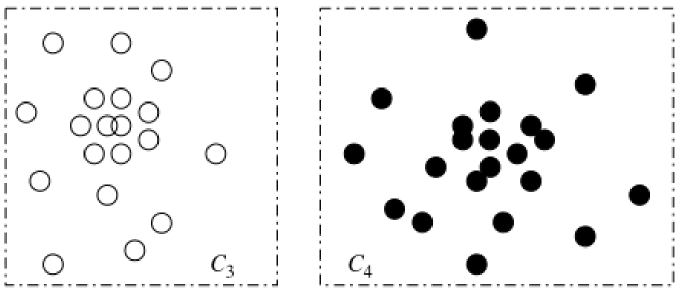
$$P(x_i|\mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x_i-\mu)^2}{2\sigma^2}}.$$

10.3.5 Probabilistic Hierarchical Clustering

- 예시



(a)



(b) (c)

$$
\begin{aligned}
&Q((\{C_1,\ldots,C_m\} - \{C_{j_1},C_{j_2}\}) \cup \{C_{j_1} \cup C_{j_2}\}) - Q(\{C_1,\ldots,C_m\}) \\
&= \frac{\prod_{i=1}^{m} P(C_i) \cdot P(C_{j_1} \cup C_{j_2})}{P(C_{j_1})P(C_{j_2})} - \prod_{i=1}^{m} P(C_i) \\
&= \prod_{i=1}^{m} P(C_i) \left(\frac{P(C_{j_1} \cup C_{j_2})}{P(C_{j_1})P(C_{j_2})} - 1 \right).
\end{aligned}
$$

10.3.5 Probabilistic Hierarchical Clustering

- 알고리즘

(1) **create** a cluster for each object $C_i = \{\boldsymbol{o}_i\}, 1 \le i \le n$;

(2) **for** $i = 1$ to n

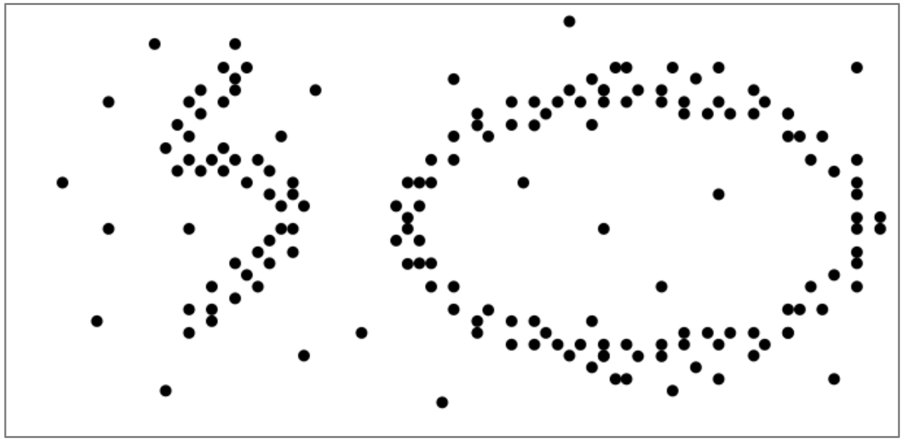
(3) **find** pair of clusters C_i and C_j such that $C_i, C_j = \arg\max_{i \neq j} \log \frac{P(C_i \cup C_j)}{P(C_i)P(C_j)}$;

(4) **if** $\log \frac{P(C_i \cup C_j)}{P(C_i)P(C_j)} > 0$ then merge C_i and C_j;

(5) **else** stop;

10.4 Density-Based Methods

- 분할과 구조적 클러스터링 알고리즘은 구 형태의 클러스터를 찾기 위해 만들어졌다.



- 밀도 기반 클러스터링을 통해 구가 아닌 형태의 클러스터를 발견.
- 종류: DBSCAN, OPTICS, DENCLUE

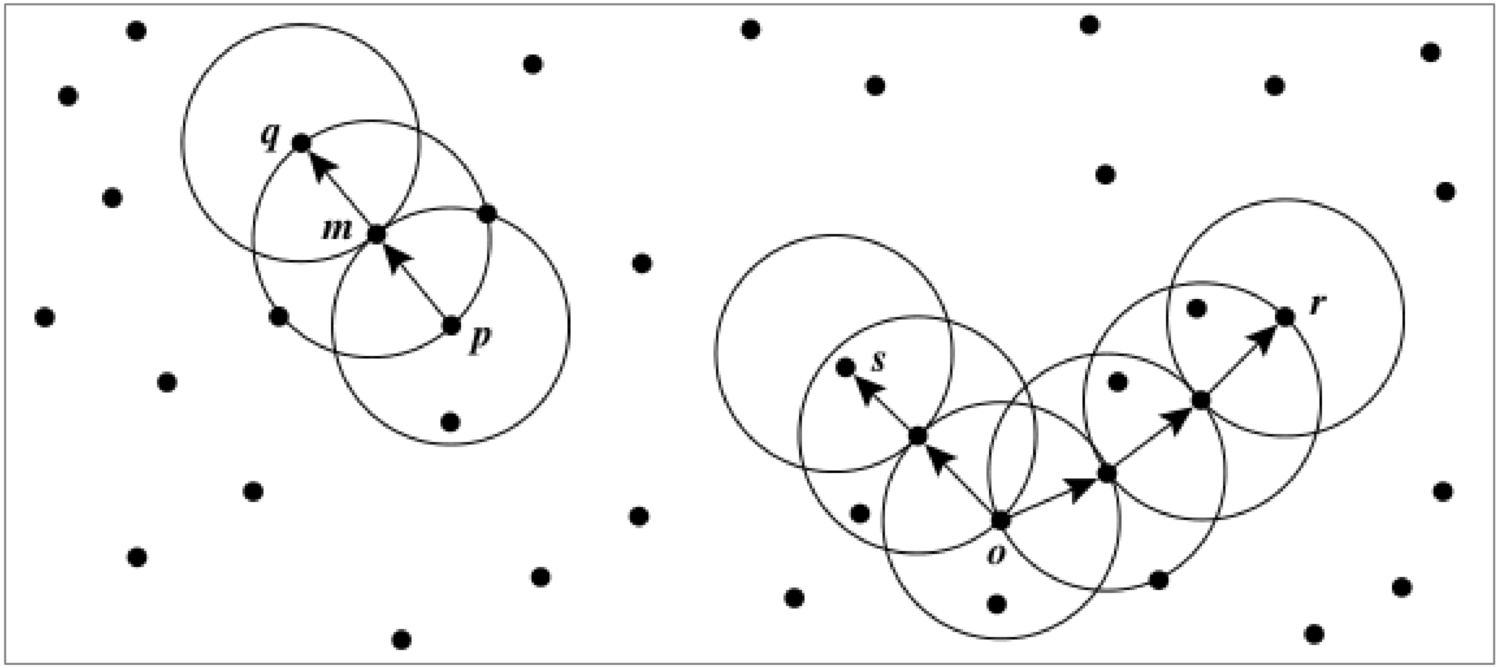
10.4.1 DBSCAN: Density-Based Clustering Based on Connected Regions with High Density

- 오브젝트 o의 밀도는 o에 인접한 오브젝트 개수로 규정 가능.
- 원의 반경 : ϵ , 반경 내의 최소 오브젝트 개수 : MinPts.
- ϵ-이웃 공간 안에 최소 MinPts개 이상의 오브젝트가 존재하는 오브젝트를 '핵심 오브젝트' 로 정의.

10.4.1 DBSCAN: Density-Based Clustering Based on Connected Regions with High Density

- 예

  - 원의 반경 : ∈ , MinPts: 3



10.4.1 DBSCAN: Density-Based Clustering Based on Connected Regions with High Density

Method:

(1) mark all objects as unvisited;
(2) **do**
(3)   randomly select an unvisited object p;
(4)   mark p as visited;
(5)   **if** the ϵ-neighborhood of p has at least *MinPts* objects
(6)   create a new cluster C, and add p to C;
(7)   let N be the set of objects in the ϵ-neighborhood of p;
(8)   **for** each point p' in N
(9)   if p' is unvisited
(10)   mark p' as visited;
(11)   if the ϵ-neighborhood of p' has at least *MinPts* points,
                add those points to N;
(12)   if p' is not yet a member of any cluster, add p' to C;
(13)   **end for**
(14)   output C;
(15)   **else** mark p as noise;
(16) **until** no object is unvisited;

10.4.1 DBSCAN: Density-Based Clustering Based on Connected Regions with High Density

- **DBSCAN이 동작하는 과정을 애니매이션으로 볼 수 있는 사이트**

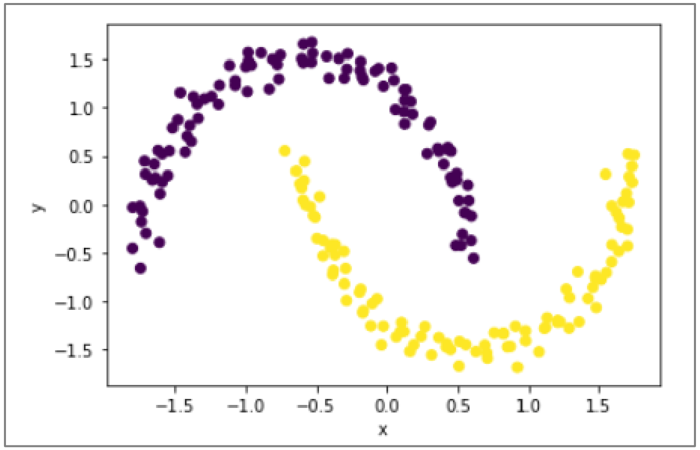
  https://www.naftaliharris.com/blog/visualizing-dbscan-clustering/

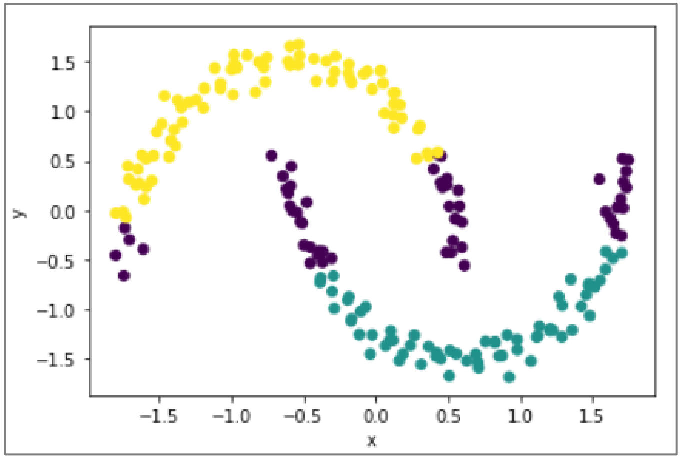
10.4.1 DBSCAN: Density-Based Clustering Based on Connected Regions with High Density

- 결과 예시

```python
# DBSCAN
from sklearn.cluster import DBSCAN
dbscan = DBSCAN(eps=0.5, min_samples=10) # 기본값
cluster = dbscan.fit_predict(scaled_X)
df["cluster"] = cluster
```

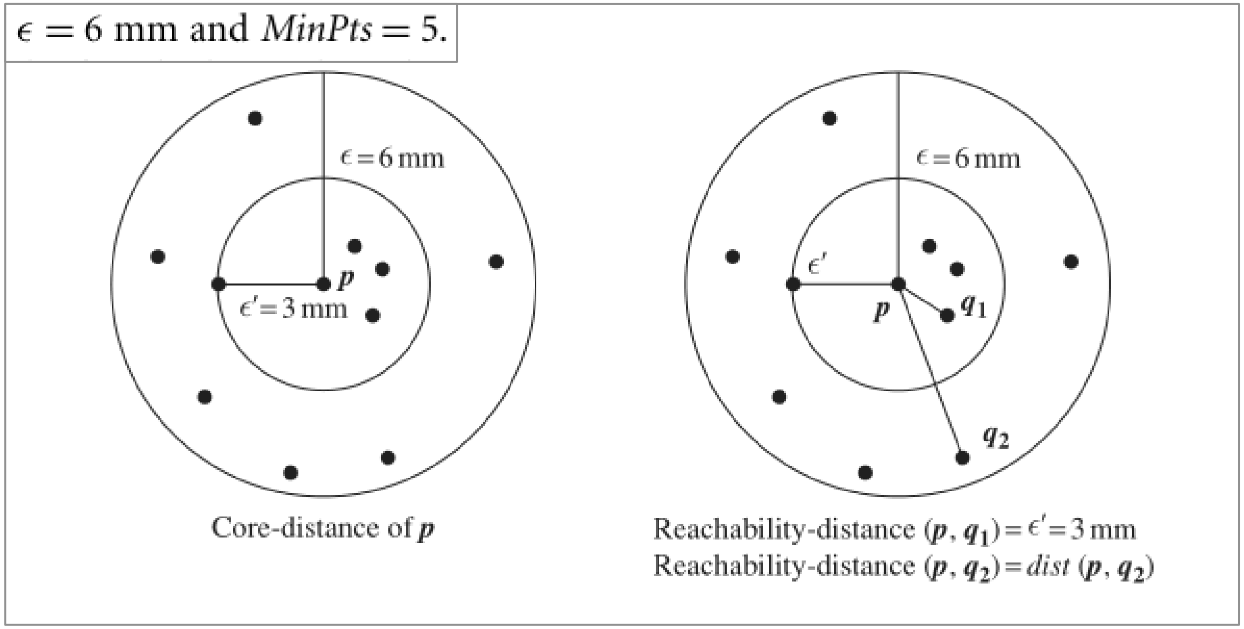
```python
# DBSCAN
from sklearn.cluster import DBSCAN
dbscan = DBSCAN(eps=0.5, min_samples=20)
cluster = dbscan.fit_predict(scaled_X)
df["cluster"] = cluster
```





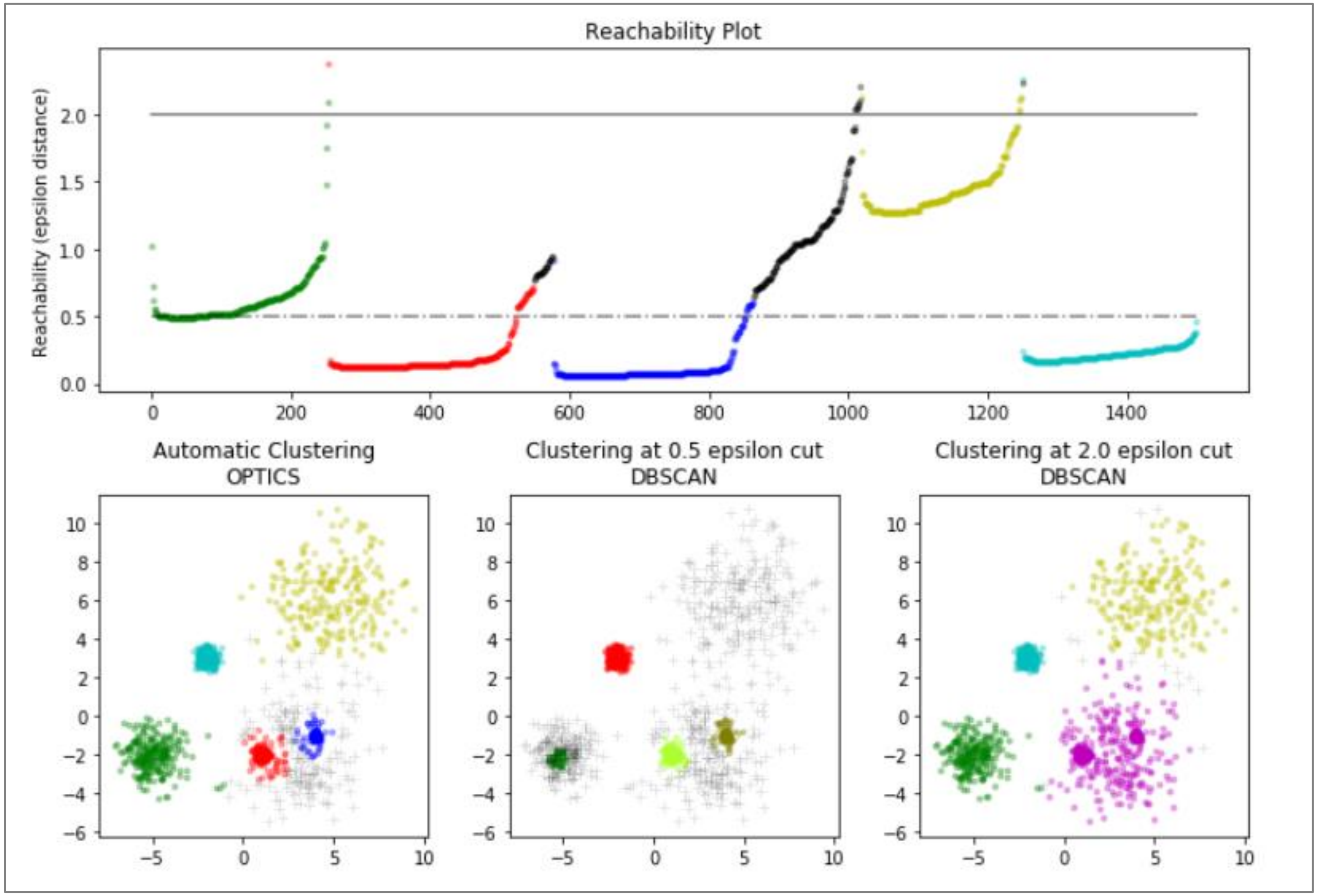
10.4.2 OPTICS: Ordering Points to Identify the Clustering Structure

- DBSCAN은 입력 파라미터에 따라 결과가 달라진다.
- 핵심 거리: 한 오브젝트 p의 핵심거리는 최소 MinPts 개의 ∈′-이웃을 만족하는 가장 작은 ∈′값을 말한다.
- 접근 거리: q에서 p의 접근 거리는 max{핵심거리(q), 거리(p, q)}가 된다.



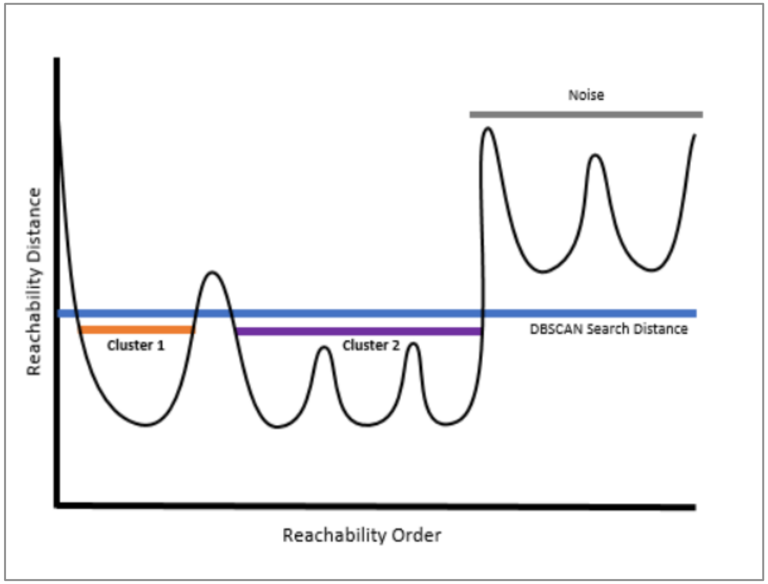
10.4.2 OPTICS: Ordering Points to Identify the Clustering Structure

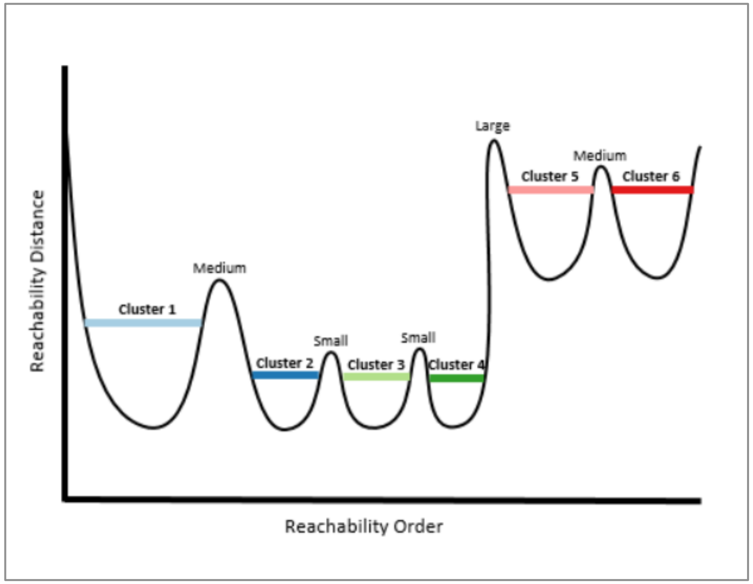
- 예시



10.4.2 OPTICS: Ordering Points to Identify the Clustering Structure

- 예시

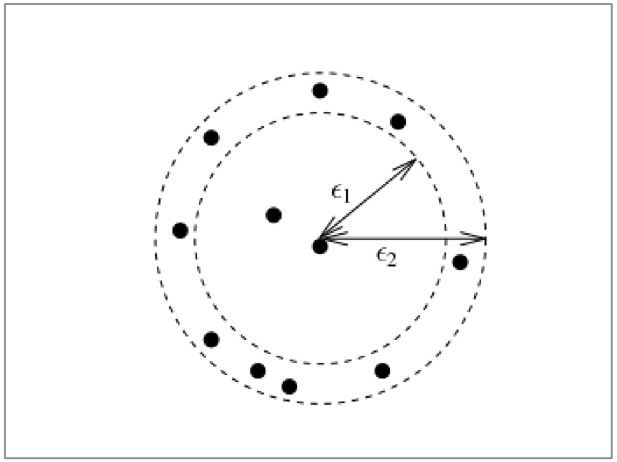




10.4.3 DENCLUE: Clustering Based on Density Distribution Functions

- 일군의 밀도 분포 함수로 클러스터를 찾아내는 클러스터링 알고리즘.

- 이웃 반경에 약간의 차이에도 밀도가 급격하게 변함.



10.4.3 DENCLUE: Clustering Based on Density Distribution Functions

- 이웃 반경에 따라 급격한 변화를 극복하기 위해 커널 밀도 추정을 사용
- 커널 밀도 추정 - 파라미터 없이 밀도를 추측할 수 있는 통계 기법

$$\hat{f}_h(x) = \frac{1}{nh}\sum_{i=1}^{n} K\left(\frac{\mathbf{x}-\mathbf{x_i}}{h}\right),$$

- 커널 함수

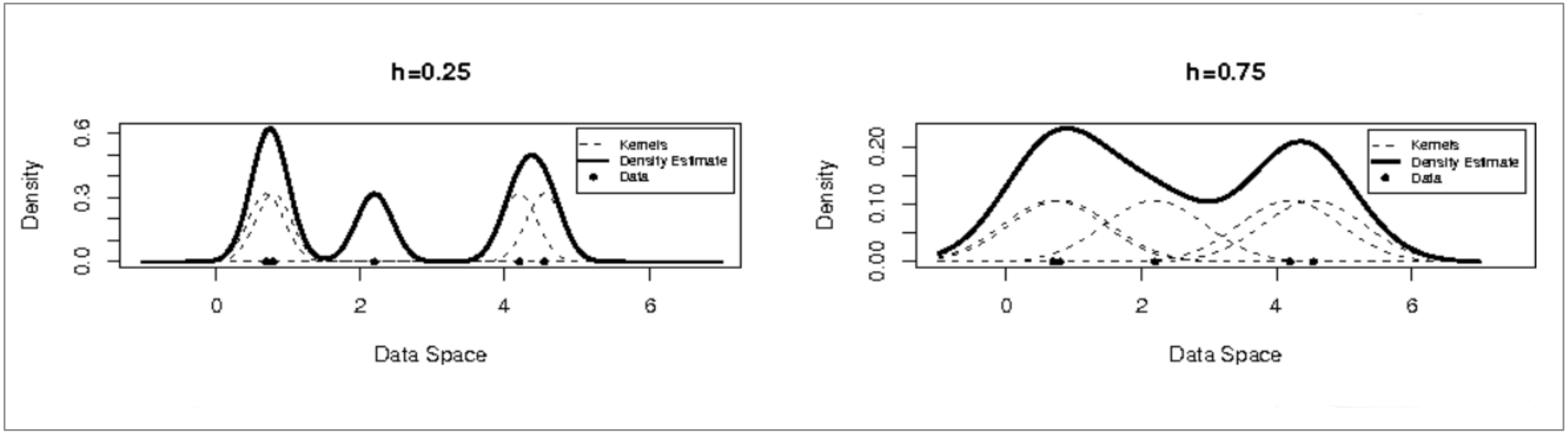
$$K\left(\frac{\mathbf{x}-\mathbf{x_i}}{h}\right) = \frac{1}{\sqrt{2\pi}} e^{-\frac{(\mathbf{x}-\mathbf{x_i})^2}{2h^2}}.$$

$$\int_{-\infty}^{\infty} K(u)du = 1 \quad \text{--- (2)}$$

$$K(u)=K(-u),\ \ K(u)\geq 0,\ \ \forall u \quad \text{--- (3)}$$

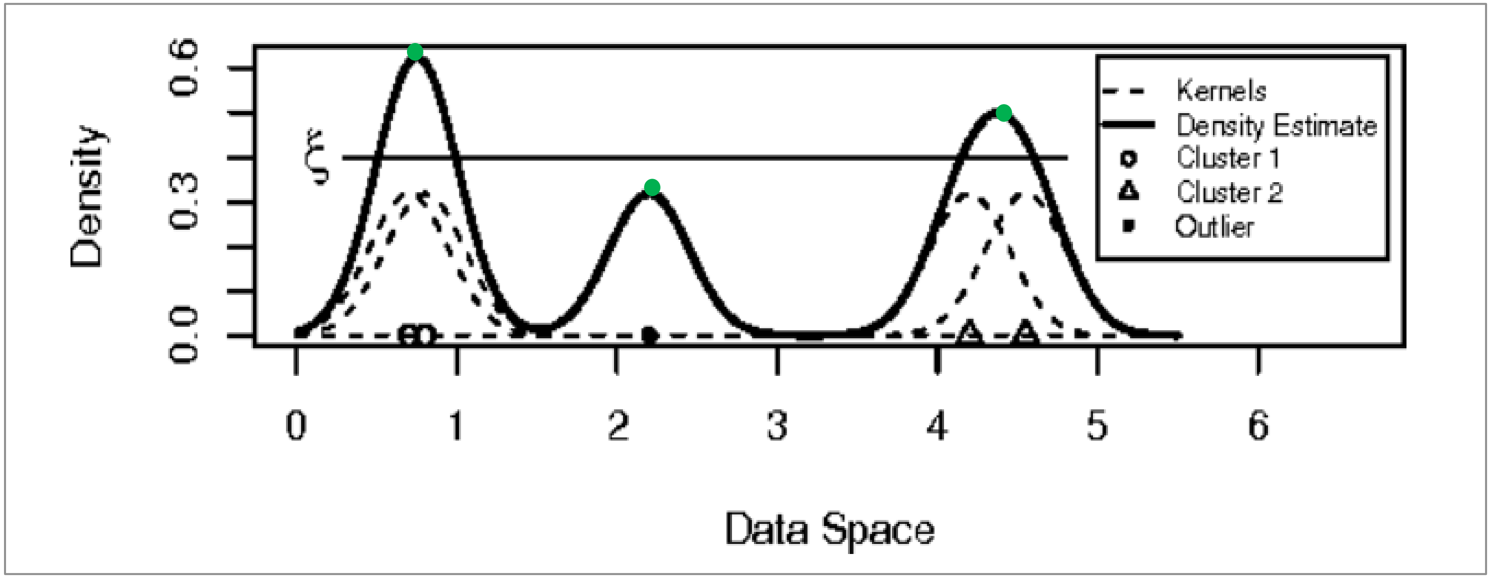
10.4.3 DENCLUE: Clustering Based on Density Distribution Functions

- h는 범위 값으로 조정 파라미터 역할.



10.4.3 DENCLUE: Clustering Based on Density Distribution Functions

- A point x′ is called a density attractor if it is a local maximum of the estimated density function.
- only considers those density attractors $\boldsymbol{x}^*$ such that $\boxed{f(\boldsymbol{x}^*) \geq \xi.}$



10.4.3 DENCLUE: Clustering Based on Density Distribution Functions

- 알고리즘
- 1. 임의의 x 선택

$$x^0 = x$$

$$x^{j+1} = x^j + \delta \frac{\nabla \hat{f}(x^j)}{|\nabla \hat{f}(x^j)|},$$

$$\nabla \hat{f}(x) = \frac{1}{h^{d+2} n \sum_{i=1}^{n} K\left(\frac{x - x_i}{h}\right)(x_i - x)}.$$

- 2. $\hat{f}(x^{k+1}) < \hat{f}(x^k)$, 을 만족할 때 까지 지속. (K > 0)
- 3. x를 $x^* = x^k$인 밀도 끌개의 클러스터로 배정.
- 4. 만약 $x^* = x^k$가 국소 최소값이면 아웃라이어다.

클러스터 분석: 기본 개념과 방법론

감사합니다