HDBSCAN

DBSCAN Overview

- 밀도 기반의 클러스터링 알고리즘으로 Core Points와 그 이웃들을 연결하여 클러스터를 형성하는 방법임.
- 특징: 데이터에 대한 분포 가정이 적고 noise를 포함함.
- 중요 개념
  - Core Points: 거리 파라미터인 ϵ 안에 이웃 점들이 최소 minPts개 이상 있는 점들
  - Border Points: Core Points는 아니지만 Core Points의 ϵ 이내에 있는 점들
  - Noise Points: Core Points도 아니고 Border points도 아닌 점들
- 한계점
  - ϵ 파라미터 값을 모든 클러스터에 적용하여 다양한 밀도를 가진 데이터에 맞지 않음.
  - 파라미터 (ϵ, minPts)를 정하기 어려움.

  ⇒ HDBSCAN은 이러한 한계를 다루며 DBSCAN을 확장함. HDBSCAN에서는 mutual reachability distance 이라는 새로운 개념 적용.

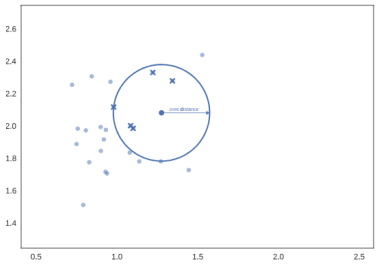
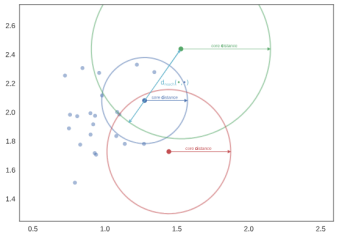
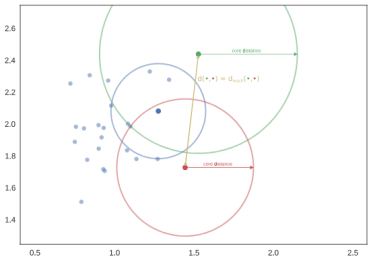
HDBSCAN (Hierarchical Density-Based Spatial Clustering of Applications with Noise)

- DBSCAN를 기반으로 계층적(Hierarchical) 개념을 더하여 개선한 클러스터링 알고리즘임.
- HDBSCAN은 거리 파라미터(ϵ)를 다양하게 적용하여 계층적 클러스터를 구성함.
- 단일 ϵ 대신, mutual reachability distance를 활용하여 모든 가능한 값들을 minimum spanning tree (MST)를 만듬.
- 중요 개념
  - **Mutual Reachability Distance**
    - 두 점 간의 대칭적인 접근을 측정하는 방식
    - p의 core distance*, q의 core distance, p와 q의 실제 거리 중 가장 큰 값으로 정의됨.

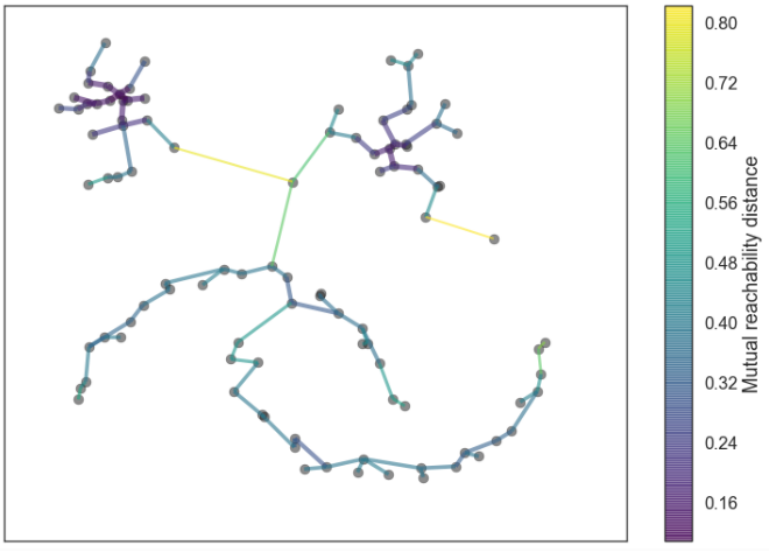
* core distance: **'minPts'**(이웃 점들의 최소 개수 k개)를 충족하는 거리(반경)

the
*k*th nearest neighbor.

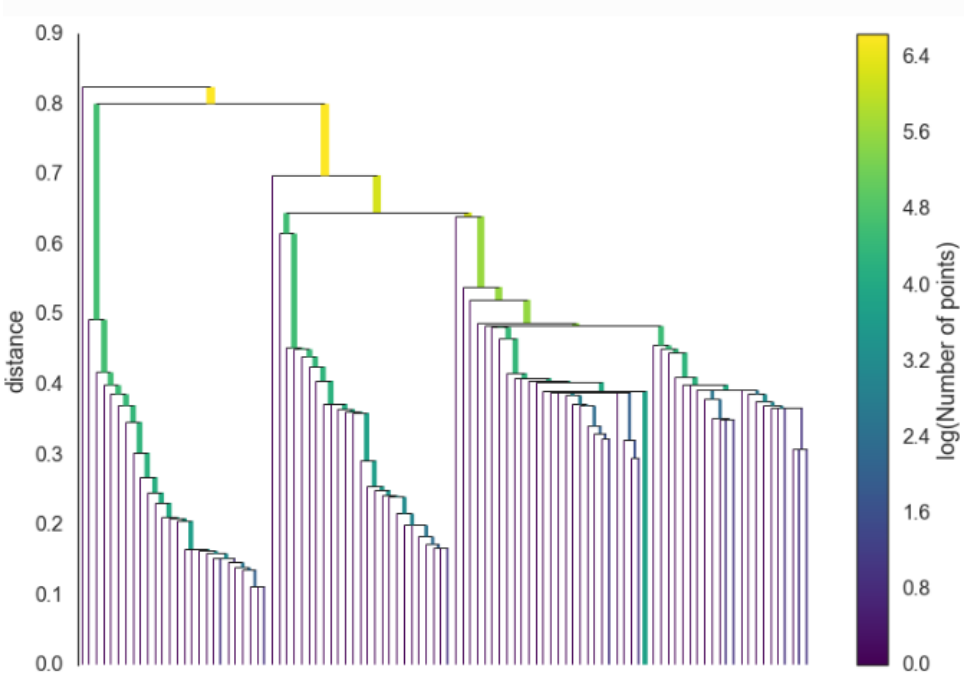
$$d_{\text{mreach-k}}(a,b) = \max\{\text{core}_k(a), \text{core}_k(b), d(a,b)\}$$

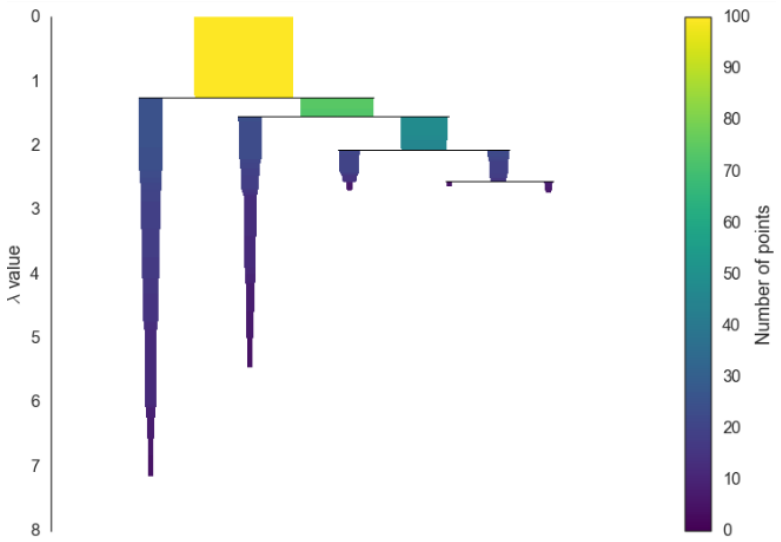
-
  - **Minimum Spanning Tree(MST)**
    - 주어진 그래프의 모든 점들을 연결하는 최소 비용을 표현한 트리(그래프 이론에서 사용되는 개념)
    - **그래프 구성**: 각 노드는 데이터 포인트를 나타내고 간선의 가중치는 두 노드 간의 mutual reachability distance
    - **MST 생성**: Prim's algorithm 또는 Kruskal's algorithm과 같은 알고리즘을 사용하여 MST를 생성
    - **연결성**: MST는 모든 데이터 포인트를 포함하며, 간선의 총 가중치 합이 최소가 되도록 연결
    - 클러스터링 알고리즘에서 클러스터의 초기 구조를 파악하는 데 사용됨.
    - HDBSCAN에서는 MST를 사용하여 점들 간의 밀도 기반 연결을 형성함.



- **Condensed Tree**
  - MST를 기반으로 안정성(stability)을 평가하여 불안정한 클러스터를 제거 후 최종 클러스터링 결과를 도출
  - **안정성 계산**: MST를 기반으로 계층적 클러스터링을 수행하고, 각 클러스터의 안정성을 계산



    - 안정성은 클러스터가 다양한 λ 값에서 얼마나 오랫동안 유지되는지를 측정
  - **가지치기**: 안정성이 낮은 클러스터를 제거하여 안정성이 높은 클러스터만 남김.
  - **Condensed Tree 형성**: 가지치기 후 남은 클러스터들을 연결하여 Condensed Tree를 형성함.



- **밀도와 클러스터의 안정성**
  - **밀도(λ) 파라미터**
    - 클러스터의 형성과 지속성을 평가하는 데 사용되는 스케일 파라미터
    - $\lambda = \frac{1}{\text{distance}}$
      - λ가 크다 = 거리가 가깝다 = 밀도가 높다 ⇒ 클러스터 형성
      - λ가 작다 = 거리가 멀다 = 밀도가 낮다 ⇒ 클러스터 X
    - λ는 클러스터의 안정성을 판단하는데 사용됨.
  - **클러스터의 안정성(Stability)**
    - 안정성은 클러스터가 다양한 λ값에 걸쳐 얼마나 오래 지속되는지 의미함.
      = 클러스터가 넓은
      λ 범위에 존재한다면, 클러스터가 다양한 밀도 수준에서 안정적으로 유지된다는 의미
    - $\sigma(C_i) = \sum_{x \in C_i} (\lambda_{\max, C_i}(x) - \lambda_{\min, C_i}(x))$
      ⇒ 클러스터
      C_i 에서 모든 포인트 x에 대해 λ 의 변화 범위를 합산한 값
      - $\lambda_{\max, C_i}(x) = \frac{1}{d_{mreach}(x, C_i)}$
        ⇒ 클러스터 C_i 에서 포인트 x가 속할 수 있는 최대 λ값
        - x가 클러스터C_i 안에 있다고 가정하고, x가 클러스터C_i 에서 떨어지기 시작하는 순간의 λ값

- $\lambda_{\min,C_i}(x)$

  ⇒ 클러스터 C_i 에서 포인트 x가 속할 수 있는 최소 λ값

  - x가 클러스터C_i 밖에 있다고 가정하고, x가 클러스터C_i 에 합류하는 순간의 λ값
- $\lambda_{\max,C_i}(x) - \lambda_{\min,C_i}(x)$

  ⇒ 포인트 x가 클러스터 C_i 에 속하는 λ 의 변화 범위

  - 이 범위가 넓을수록 x는 다양한 밀도 수준에서 클러스터에 속함.
- $\sum_{x \in C_i}$

  ⇒ 안정성의 합산으로 클러스터 전체의 지속성을 측정함.

  - 값이 클수록 클러스터가 다양한 λ 범위에서 안정적으로 유지된다는 것을 의미

Algorithm Steps

1. **Transform the space**
   - 데이터 포인트 간의 거리를 재정의하여 클러스터링에 적합한 공간을 만드는 과정
   - 각 포인트 간에 core distance를 구하고 Mutual Reachability Distance를 정의하여 데이터 공간을 변환.
2. **Build the minimum spanning tree**
   - Mutual Reachability Distance를 간선의 가중치로 사용하여 MST를 구축
   - MST는 주어진 노드 집합을 연결하는 최소 비용의 트리
   - 모든 이웃에 대해 거리를 구하는 것은 비용이 많이 들기 때문에 Prim's algorithm *을 이용해 효율적으로 MST를 만듦.

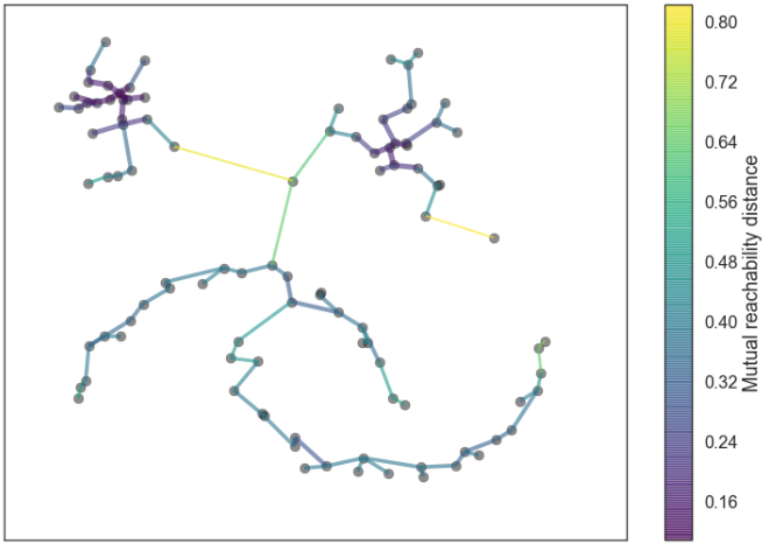
     * 프림 알고리즘: 가중치가 있는 무방향 그래프를 위해 최소 스패닝 트리를 구성하는 탐욕 알고리즘(greedy algorithm)으로 임의의 시작점에서 출발해 트리를 확장하며, 각 단계에서 현재 트리에 가장 가까운 가중치가 작은 점을 선택함.

     (순서)

     1) 임의의 시작점에서 시작

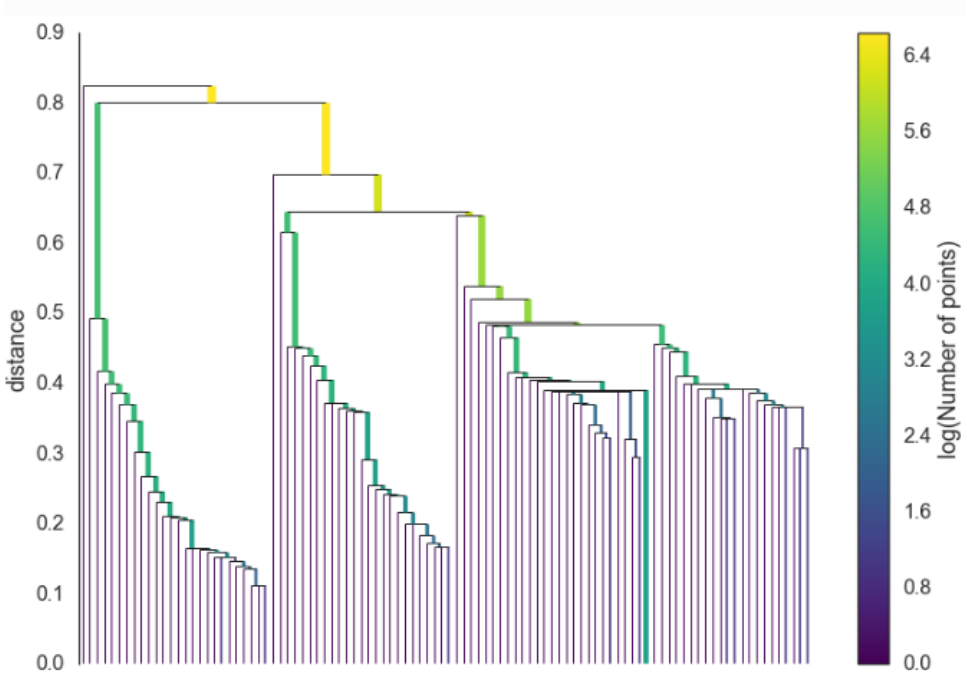
2) 현재 트리에 포함된 점과 포함되지 않은 점 간의 가장 작은 가중치를 가진 선을 반복적으로 선택하여 트리에 추가합니다.

2) 모든 정점이 MST에 포함될 때까지 이 과정을 반복합니다.



3. **Build the cluster hierarchy**

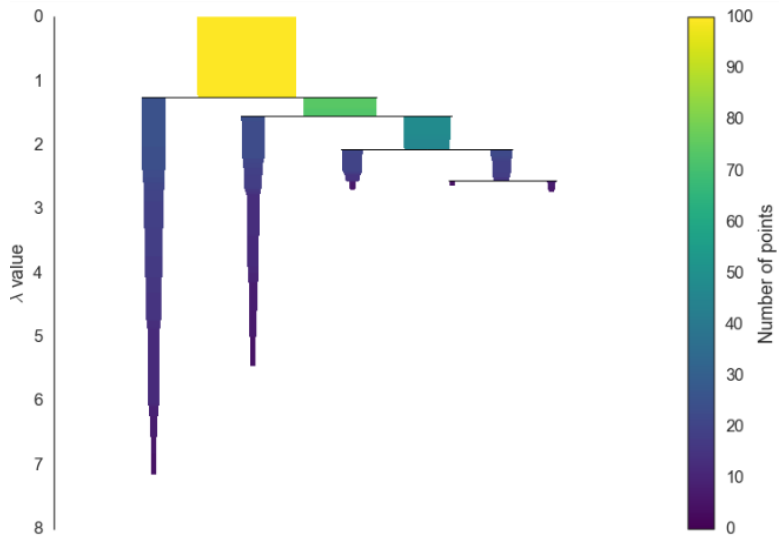
- MST가 주어지면 연결된 구성 요소의 계층 구조로 변환하는 과정



- MST의 연결선을 거리(Mutual Reachability Distance) 순으로 오름차순 정렬 후, 클러스터를 병합하여 계층 구조 생성

- 덴드로그램을 통해 클러스터의 계층 구조를 시각화하며 다양한 레벨에서 클러스터를 선택하여 가변 밀도 클러스터를 처리할 수 있음.

4. **Condense the cluster tree**
    - 클러스터 추출하기 위한 단계로 복잡한 계층 구조를 축소하여 더 작은 트리로 압축



- 클러스터 분할 시 한두개의 점들이 클러스터에서 분리될 때, 단일 클러스터가 포인트를 잃는 것으로 봄. 이를 구체화 하기 위해 **minimum cluster size(최소 클러스터 개수)** 파라미터가 사용됨.
- 계층 구조를 통해 분할을 탐색하면서 생성된 클러스터 각각이 **minimum cluster size보다 작은지 확인**
- 새로 생성된 클러스터 크기 < **minimum cluster size**
  →
  클러스터에서 떨어져 나간 포인트로 간주 & 더 큰 클러스터는 부모 클러스터의 정체성을 유지
- 새로 생성된 클러스터 크기 > **minimum cluster size**

  → 실제 분할로 간주하고 트리에 그 분할을 유지시킴.

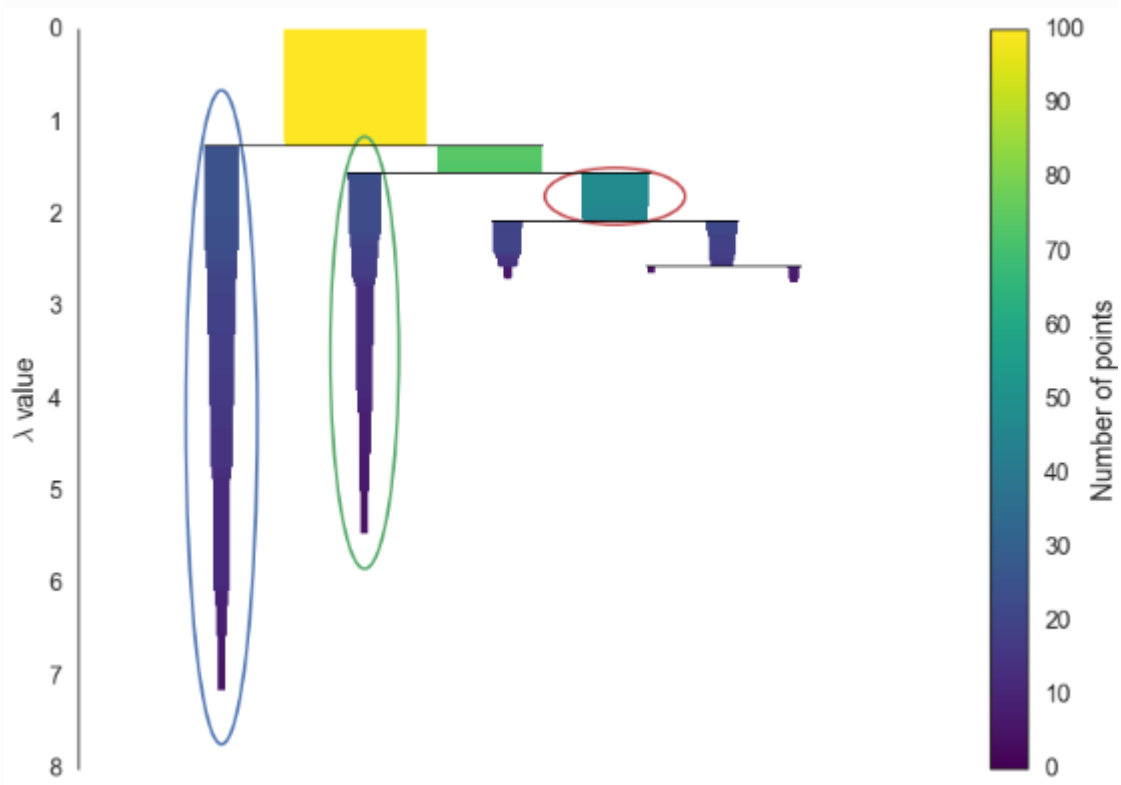
  * $\lambda = \frac{1}{\text{distance}}$

5. **Extract the clusters**
    - 기준 밀도의 변동이 있어도 안정적으로 오랫동안 존재하는 클러스터를 선택하자. (면적이 큰 클러스터)

- 제약: 특정 클러스터를 선택하면 그 자손 클러스터는 선택할 수 없음.

1. 안정성 계산
   - $\sigma(C_i) = \sum_{x \in C_i} (\lambda_{\max,C_i}(x) - \lambda_{\min,C_i}(x))$
2. 리프노드 선택
   - 처음에는모든 리프노드를 선택된 클러스터로 선언
3. 역순 탐색
   - 트리의 루트 노드까지 거꾸로 올라가며 작업
   - 자식 클러스터 안정성 합과 부모 클러스터 안정성 비교
     - 자식 클러스터의 안정성 합이 더 큰 경우
       - 자식 클러스터들이 개별적으로 더 의미가 있다고 판단
     - 부모 클러스터의 안정성 합이 더 큰 경우
       - 부모 클러스터가 더 큰 의미를 가지며, 자식 클러스터들은 선택 해제

⇒ 단순히 '가장 큰 총 잉크 면적을 가진 클러스터를 선택하되 자손 제약을 고려하여 선택'하는 과정



- 선택되지 않은 클러스터에 속하지 않는 모든 포인트는 단순히 노이즈 포인트로 간주되며 레이블 -1 할당

Reference

- Accelerated Hierarchical Density Clustering: https://arxiv.org/pdf/1705.07321v2.pdf
- https://hdbscan.readthedocs.io/en/latest/index.html
- ChatGPT

Accelerated Hierarchical Density Clustering_2017 [gpt_summary]