9.27 hierarchical clustering

2021年9月27日 星期一 下午4:41

- Two main type of hierarchical clustering
  - Agglomerative
    - Alg:
      - Let each point in the dataset be in its own cluster
      - Compute the distance between clusters
      - Merge two closet
      - Repeat
    - —> how to calculate the distance between cluster?
      - Single-link distance
        - The minimum of all pairwise distances between a point from one cluster and a point from the other cluster

$$D_{SL}(C_1, C_2) = \min\{d(p_1, p_2) \mid p_1 \in C_1, p_2 \in C_2\}$$

        - Pro
          - Can handle clusters of different sizes
        - Cons
          - Sensitive to noise points
          - Tends to create elongated cluster
      - Average-link distance
        - The average of all pairwise distances between a point from one cluster and a point from the other cluster
        - $$D_{AL}(C_1, C_2) = \frac{1}{|C_1| \cdot |C_2|} \sum_{p_1 \in C_1, p_2 \in C_2} d(p_1, p_2)$$
        - Pros
          - Less susceptible to noise and outliers
        - Cons
          - Tends to be biased towards globular clusters
      - Centroid distance
        - The distance between the centroid of clusters
        - $$D_C(C_1, C_2) = d(\mu_1, \mu_2)$$
      - Wards's distance
        - The difference between the spread/variance of points in the merged cluster and the unmerited clusters
        - $$D_{WD}(C_1, C_2) = \sum_{p \in C_{12}} d(p, \mu_{12}) - \sum_{p_1 \in C_1} d(p_1, \mu_1) - \sum_{p_2 \in C_2} d(p_2, \mu_2)$$
  - Divisive
    - How?
      - Start with every point in the same cluster
      - At each step, split until every point is in its own cluster

- Density based clustering
  - Cluster together points that are densely packed together
    - How to define density?
      - Given a fixed radius s around a point, if there's at least min_pts number of points in that area, then the section is dense.
  - Core point:
    - Have so much dense
  - Border point
    - Don't have so many dense point
  - Noise point
  - Neither a core nor a border
  - Algorithm:
    - **DBScan Algorithm**

      ε and **min_pts** given:

      1. Find the **ε**-neighborhood of each point
      2. Label the point as **core** if it contains at least **min_pts**
      3. Label points in its neighborhood that are not **core** as **border**
      4. Label points as **noise** if they are neither **core** nor **border**
      5. For each **core** point, assign to the same cluster all **core** points in its neighborhood
      6. Assign border points to nearby clusters
  - Pro:
    - Can identify clusters of different shapes and sizes
    - Resistant to noise
  - Cons:
    - Fail to identify clusters of varying densities
    - Tends to create clusters of the same density
    - Notion of density is problematic in high dementia spaces