CSIT5210

Other Clustering Techniques

Prepared by Raymond Wong
Presented by Raymond Wong
raywong@cse

What we learnt

- K-mean
- Dendrogram

Other Clustering Models

- Model-Based Clustering
  - EM Algorithm
- Density-Based Clustering
  - DBSCAN
- Scalable Clustering Method
  - BIRCH

EM Algorithm

- Drawback of the K-means/Dendrogram
  - Each point belongs to a single cluster
  - There is no representation that a point can belong to different clusters with different probabilities
- Use probability density to associate to each point

EM Algorithm

- Assume that we know there are k clusters
- Each cluster follows a distribution (e.g., Gaussian Distribution)
  - 1D Gaussian Distribution
    - Mean μ
    - Standard derivation σ

p(x| <μ, σ>) = $$\frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

EM Algorithm

- Since there are k clusters, we have k distributions.
- Cluster 1
  - Gaussian Distribution
    - Mean μ_1
    - Standard derivation σ_1
- Cluster 2
  - Gaussian Distribution
    - Mean μ_2
    - Standard derivation σ_2
- …
- Cluster k
  - Gaussian Distribution
    - Mean μ_k
    - Standard derivation σ_k

EM Algorithm

- EM Algorithm
  - Expectation-Maximization
- Algorithm
  - **Step 1 (Parameter Initialization)**
    - Initialize all μ_i and σ_i
  - **Step 2 (Expectation)**
    - For each point x,
      - For each cluster i,
        - Calculate the probability that x belongs to cluster i
  - **Step 3 (Maximization)**
    - For each cluster i,
      - Calculate the mean μ_i according to the probabilities that all points belong to cluster i
  - Repeat Step 2 and Step 3 until the parameters converge

One possible implementation:

$$p(x \in C_i) = \frac{p(x | <\mu_i, \sigma_i>)}{\sum_j p(x | <\mu_j, \sigma_j>)}$$

One possible implementation:

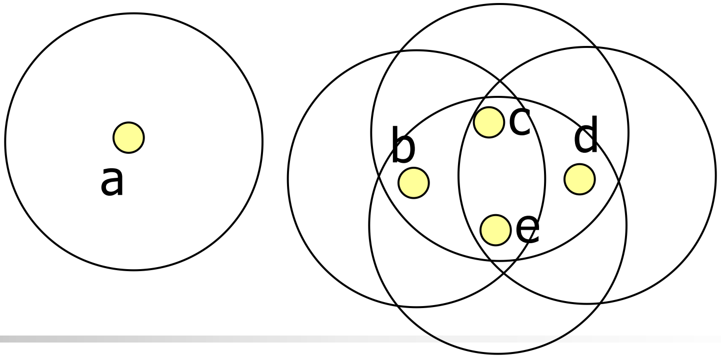
$$\mu_i = \sum_x x \cdot \frac{p(x \in C_i)}{\sum_y p(y \in C_i)}$$

Other Clustering Models

- Model-Based Clustering
  - EM Algorithm
- Density-Based Clustering
  - DBSCAN
- Scalable Clustering Method
  - BIRCH

DBSCAN

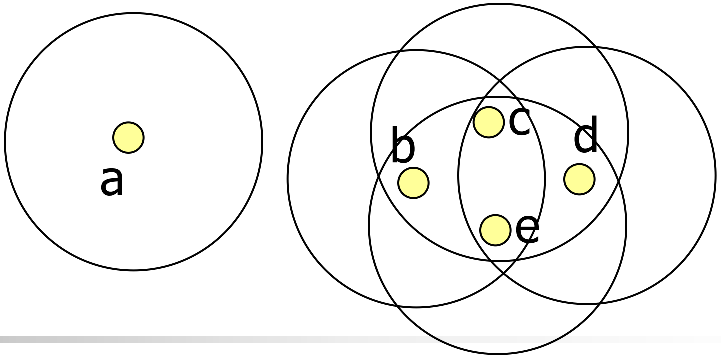
- Traditional Clustering
  - Can only represent sphere clusters
  - Cannot handle irregular shaped clusters
- DBSCAN
  - Density-Based Spatial Clustering of Applications with Noise



DBSCAN

- Given a point p and a non-negative real number ε,
  - the ***ε-neighborhood*** of point p, denoted by N(p), is the set of points q (including point p itself) such that the distance between p and q is within ε.

DBSCAN



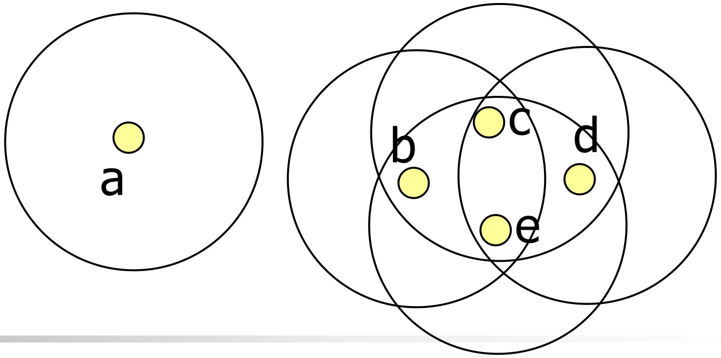
- According to ε-neighborhood of point p, we classify all points into three types
  - **core points**
  - **border points**
  - **noise points**

Given a point p and a non-negative integer MinPts, if the size of N(p) is at least MinPts, then p is said to be a **core point**.

Given a point p, p is said to be a **border point** if it is not a core point but N(p) contains at least one core point.

Given a point p, p is said to be a **noise point** if it is neither a core point nor a border point.

DBSCAN



- **Principle 1:** Each cluster contains at least one core point.
- **Principle 2:** Given any two core points p and q, if N(p) contains q (or N(q) contains p), then p and q are in the same cluster.
- **Principle 3:** Consider a border point p to be assigned to one of the clusters formed by Principle 1 and Principle 2. Suppose N(p) contains multiple core points. A border point p is assigned arbitrarily to one of the clusters containing these core points (formed by Principle 1 and Principle 2).
- **Principle 4:** All noise points do not belong to any clusters.

Other Clustering Models

- Model-Based Clustering
  - EM Algorithm
- Density-Based Clustering
  - DBSCAN
- Scalable Clustering Method
  - BIRCH

BIRCH

- Disadvantage of Previous Algorithms
  - Most previous algorithms cannot handle update
  - Most previous algorithms are not scalable
- BIRCH
  - Balanced Iterative Reducing and Clustering Using Hierarchies

BIRCH

- Advantages
  - Incremental
  - Scalable

BIRCH

- Each cluster has the following three terms.
  - Mean

$$\mu = \frac{\sum_{i=1}^{n} x_i}{n}$$

  - Radius

$$R = \sqrt{\frac{\sum_{i=1}^{n} (x_i - \mu)^2}{n}}$$

Average distance from member objects to the mean

  - Diameter

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

Average pairwise distance within a cluster

BIRCH

- L is to store a list of clusters
- Idea of the Algorithm
  - L $\leftarrow$ {}
  - When there is a new data point x
    - If L = {}
      - Create cluster C containing x only
      - Insert C into L
    - Else
      - Find the closest cluster C from x
      - Insert x into C
      - If C has diameter D greater than a given threshold,
        - Split cluster C into two sub-clusters C_1 and C_2
        - Remove C from L
        - Insert C_1 and C_2 into L

BIRCH

- If there is no efficient data structure,
  - the computation of D is very slow
  - Running time = O(?)

BIRCH

- BIRCH stores the following for each cluster instead of μ, R and D
  - clustering feature (CF)
    - <n, LS, SS>
      - n: no. of points in the cluster
      - LS: the linear sum of the n points
      - SS: the square sum of the n points

LS = $\sum_{i=1}^{n} x_i$

SS = $\sum_{i=1}^{n} x_i^{\,2}$

BIRCH

- It is easy to verify that R and D can be derived from CF

BIRCH

- Why Efficient?
- When a new data point comes, we can update the CF structure efficiently
  - Running time = O(?)
  - clustering feature (CF)
    - <n, LS, SS>
      - n: no. of points in the cluster
      - LS: the linear sum of the n points
      - SS: the square sum of the n points

$$LS = \sum_{i=1}^{n} x_i$$

$$SS = \sum_{i=1}^{n} x_i^{\ 2}$$