Chapter 7. Cluster Analysis

Dong-Kyu Chae

PI of the Data Intelligence Lab @HYU
Department of Computer Science & Data Science
Hanyang University

Data
Intelligence
LAB

Comments on the *K-Means* Method

- **Benefits:** Relatively efficient : $O(n \cdot \cancel{k} \cdot \cancel{t})$, where *n* is # objects, *k* is # clusters, and *t* is # iterations. Normally, *k*, *t* << *n*

  can ignore

  - Comparing: PAM: $O(k(n-k)^2)$, CLARA: $O(ks^2 + k(n-k))$

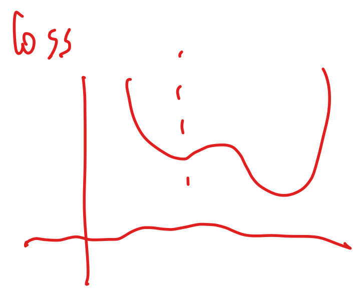
    clustering of the entire data

    # of data in a sampled DB

    PAM

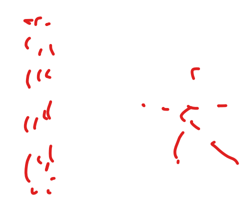
- **Comment:** it may terminate at a local optimum

  starting point가 randomly 해서



- **Limitations**
  - Programmers need to specify k (the number of clusters)
  - Unable to handle noises and outliers
  - Not suitable to discover clusters with non-convex shapes

    볼록하지 않은



이런거에 적합 X

Variations of the *K-Means* Method

only for numerical type

categorical data의 average 구하기 힘듦

student ceo...

- **Handling categorical-only data: *k-modes***

k-means 와 difference 한계 2개있다

  - X, Y: objects having *m* categorical features
  - ① Distance d(X,Y): the number of total mismatched features

$$d(X,Y)=\Sigma_{j=1}^{m}\delta(x_j,y_j) \quad \text{where} \quad \delta(x_j,y_j)=\begin{cases}0 & (x_j = y_j)\\ 1 & (x_j \neq y_j)\end{cases}$$

  - ② Mode of each cluster $K_1, K_2, \ldots, K_k$ is a vector $Q = \langle q_1, q_2, \ldots, q_m \rangle$ that minimizes

$$D(X,Q)=\Sigma_{i=1}^{n}d(X_i,Q)$$

student

minimize 하고싶다

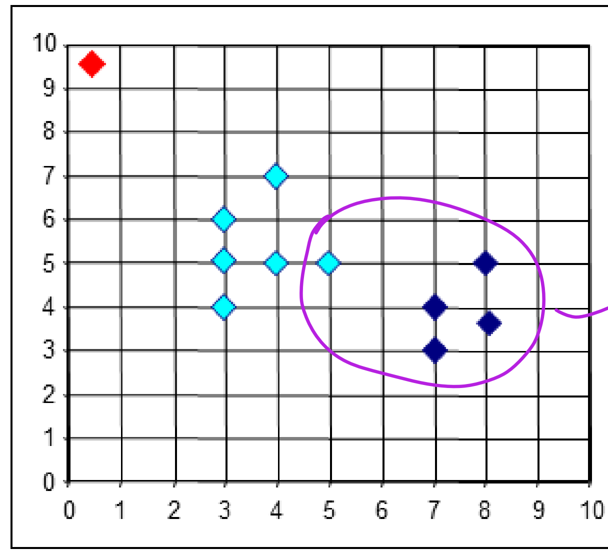
  - Finding each mode Q
    - Taking the value **most frequently occurring** for each feature
- **A mixture of categorical and numerical data: *k-prototype* method (skipped)**

Problem of K-Means Clustering

- **The k-means algorithm is sensitive to outliers**
  - An object with an extremely large value may substantially distort the distribution of the data

1. 먼저 detection 하고 remove 하고

2. k-medoid 사용



outlier 때문에 이렇게 될 수 있다

- **K-Medoids:** Instead of taking the mean value (i.e., *centroids*) of the object in a cluster as a reference point, a medoid can be used, which is the most centrally-located object in a cluster

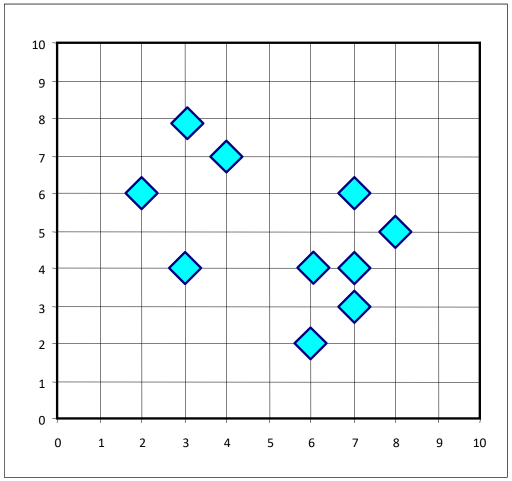
real object를 택함

PAM (Partitioning Around Medoids)

- **PAM is a typical K-Medoids algorithm**
- **Use a real object to represent the cluster**
  1. Randomly select ***k*** seed objects
  2. For each pair of selected seed ***i*** and non-seed object ***h***, calculate the **total swapping cost** TC_{ih}
     - It measures the quality of clustering before/after swapping the role of *i* and *h*.

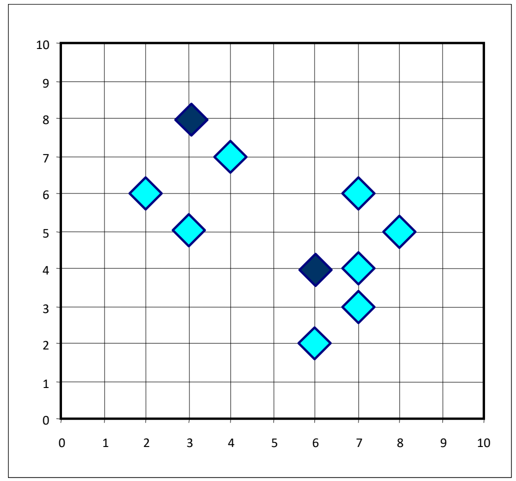
       d(after) − d(before)
  3. For each pair of ***i*** and ***h***,
     - If $TC_{ih} < 0$, ***i*** is replaced by ***h***   d(before)이 더 크니까 h를 new seed로
     - Then, each non-selected object is assigned to the most similar seed
  4. Repeat steps 2-3 until there is no change

PAM: Algorithm Overview

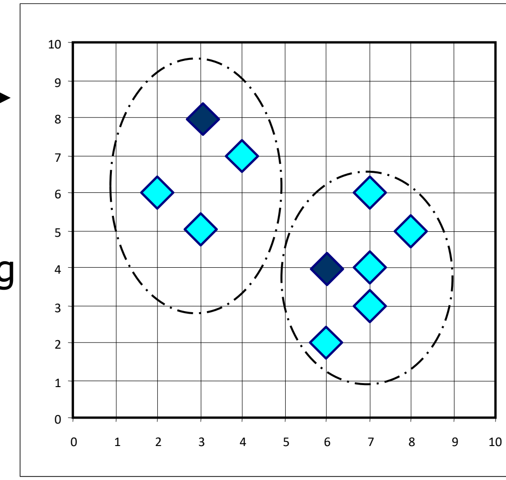


K=2

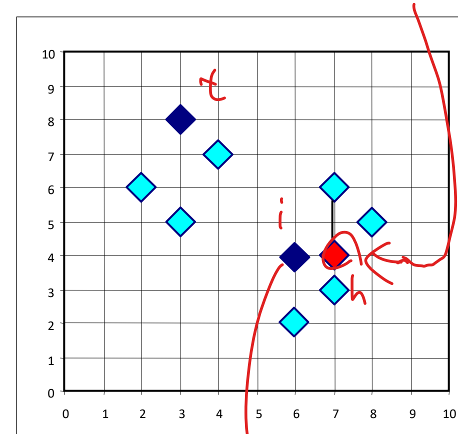
Randomly choose k seeds as initial medoids



Assign each remaining object to nearest medoids



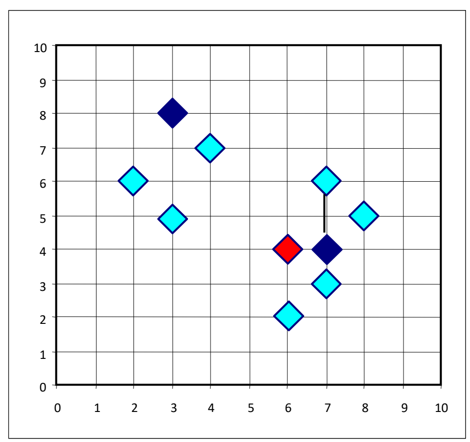
Randomly select a nonmedoid object, O_{ramdom}



m이 안으로 계산하면

8-10=-2 같은 clusta에서 10
h는 8

Compute **total swapping cost**



reassign

Swap O and O_{ramdom}

If quality is **improved.**

Do loop

Until no change

Total Swapping Cost $TC_{ih}=\sum_j C_{jih}$

C_{jih} =d(new distance) – d(old distance)
=d(j, seed_after) – d(j, seed_before)

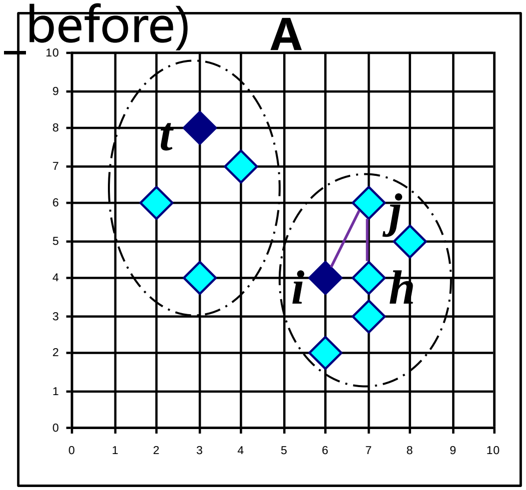
i: original seed
h: new seed
t: other seed
j: non-seed

A: j belonged to i and now belongs to h

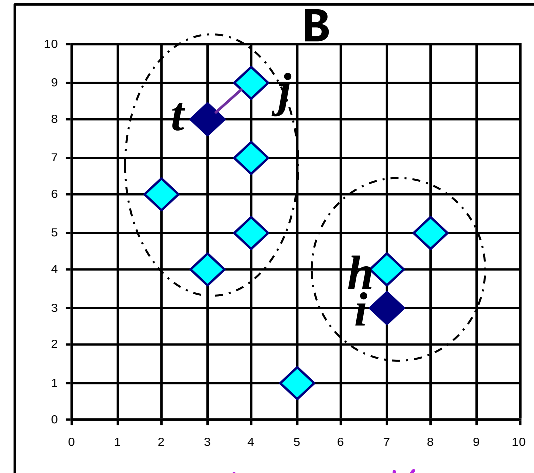
B: j belonged to t and again belongs to t   j does not care

C: j belonged to i and now belongs to t

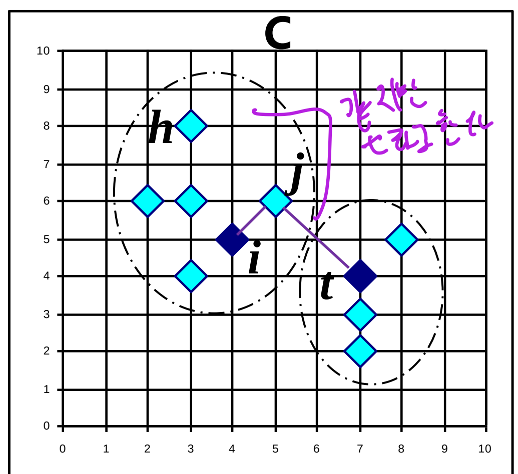
D: j belonged to t and now belongs to h



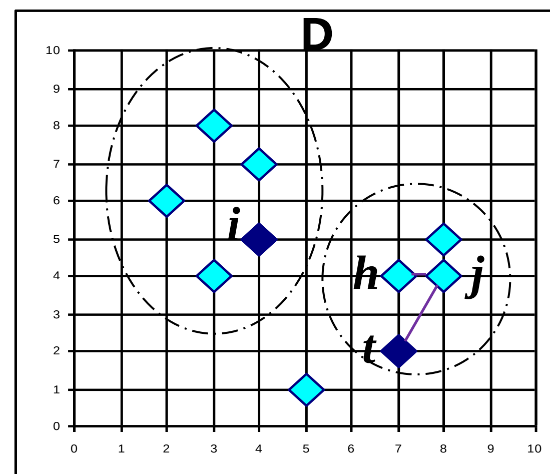
C_{jih} = d(j, h) - d(j, i)



C_{jih} = 0 d(j,t) - d(j,t)



C_{jih} = d(j, t) - d(j, i)



C_{jih} = d(j, h) - d(j, t)

PAM (Partitioning Around Medoids)

- **PAM is more robust than k-means in the presence of noise and outliers**
  - because a medoid is less influenced by outliers or other extreme values than a mean (i.e., centroid)
- **PAM works efficiently for small data sets but does not scale well for large data sets.**
  - $O(i*k*\underbrace{(n-k)^2}_{n^2})$ where *n* is # of data, *k* is # of clusters, *i* is # of iterations

➔**Solution:** sampling-based approach

Example: **CLARA (Clustering LARge Applications)**

CLARA (Clustering Large Applications)

- **CLARA** draws multiple samples of the full dataset
  - For each sampled dataset, it applies *PAM* to get the medoids
  - Then the entire data is clustered based on the medoids
  - The clustering quality is then evaluated
  - Choose the medoids yielding the best qualiy of clustering
- **Strength:** deals with larger data sets than PAM
- **Weakness:**
  - Efficiency depends on the sample size
  - A good clustering based on samples will not necessarily represent a good clustering of the whole data set if the sample is biased

repeat

PAM M_1

$k = 10 \sim 100$

$(M_1, \dots, M_{10})$

$S \uparrow \rightarrow$ quality $\uparrow$, time $\uparrow$

time $\leftrightarrow$ Quality tradeoff

Thank You