CS 461
Artificial Intelligence

Dr. Hashim Yasin

Limitations in K-means Clustering

Limitations of K-means

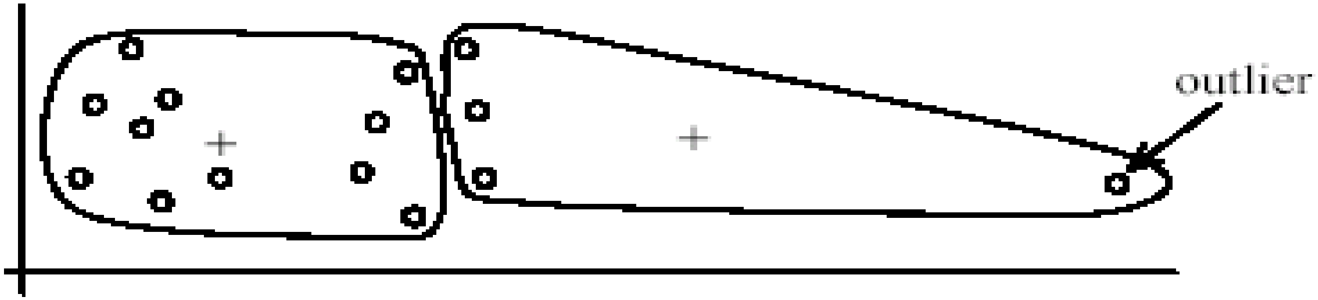
- K-means has problems when the data contains outliers

- *The K-means algorithm is very sensitive to the **initial seeds**.*

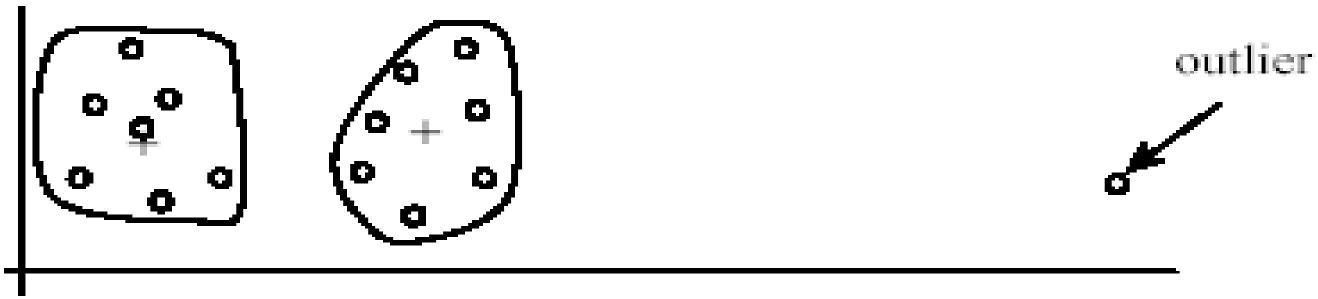
- K-means has problems when clusters are of different
  - Sizes
  - Densities
  - Non-globular shapes

Limitations of K-means

- K-means has problems when the data contains outliers



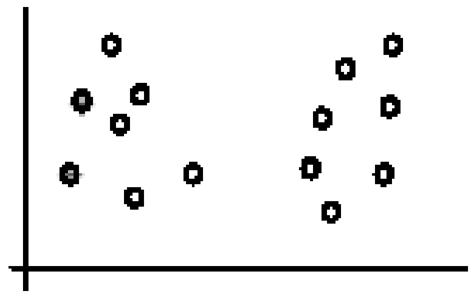
(A): Undesirable clusters



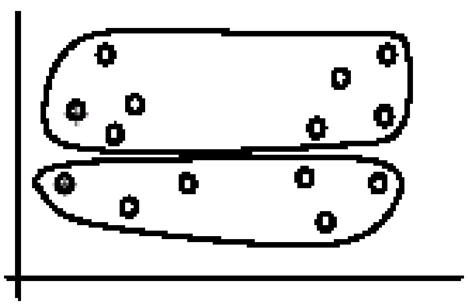
(B): Ideal clusters

Limitations of K-means

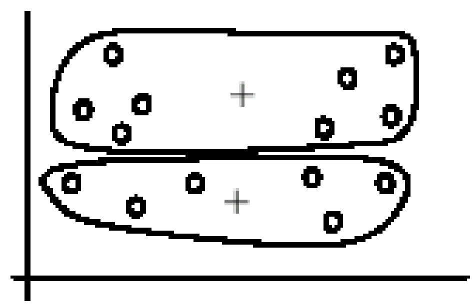
- The algorithm is sensitive to initial seeds



(A). Random selection of seeds (centroids)



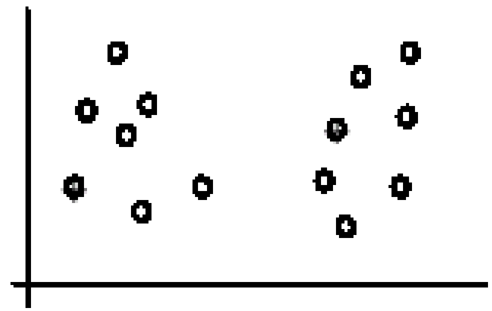
(B). Iteration 1



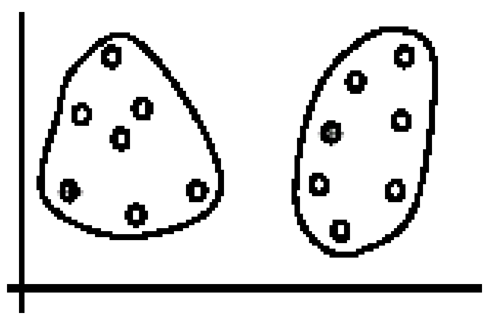
(C). Iteration 2

Limitations of K-means

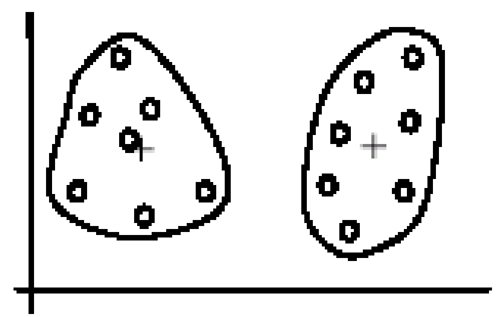
- The algorithm is sensitive to initial seeds



(A). Random selection of k seeds (centroids)



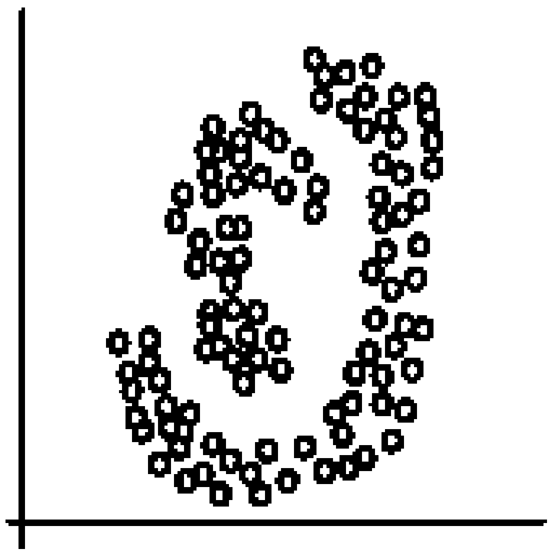
(B). Iteration 1



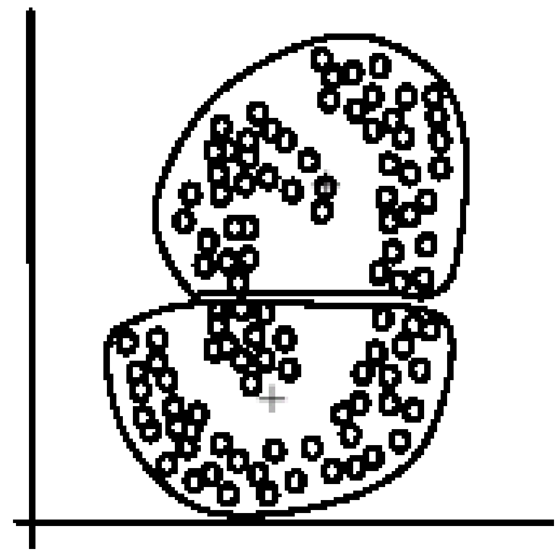
(C). Iteration 2

Limitations of K-means

- The *k*-means algorithm is not suitable for discovering clusters that are not hyper-ellipsoids (or hyper-spheres).



(A): Two natural clusters



(B): *k*-means clusters

Data Standardization

Data Standardization

- In the Euclidean space, standardization of attributes is recommended so that all attributes can have equal impact on the computation of distances.
- Consider the following pair of data points:

$$\boldsymbol{x}_i\text{: } (0.1, 20) \text{ and } \boldsymbol{x}_j\text{: } (0.9, 720)$$

$$dist(\mathbf{x}_i, \mathbf{x}_j) = \sqrt{(0.9 - 0.1)^2 + (720 - 20)^2} = 700.000457,$$

- The distance is almost completely dominated by $(720 - 20) = 700$.
- Standardize attributes: to force the attributes to have a common value range,

Data Standardization

Interval-scaled attributes:

- Their values are real numbers following a linear scale.
  - The difference in Age between 10 and 20 is the same as that between 40 and 50.
  - The key idea is that intervals keep the same importance through out the scale
- Two main approaches to standardize interval scaled attributes, **range** and **z-score**.

Range:

- Consider *f* is an attribute

$$range(x_{if}) = \frac{x_{if} - \min(f)}{\max(f) - \min(f)},$$

Data Standardization

Z-score:

- transforms the attribute values so that they have a mean of zero and a mean absolute deviation of 1.
- The mean and absolute deviation of attribute *f*, denoted by m_f and s_f respectively is computed as,

$$m_f = \frac{1}{n}\left(x_{1f} + x_{2f} + \ldots + x_{nf}\right),$$

$$s_f = \frac{1}{n}\left(|x_{1f} - m_f| + |x_{2f} - m_f| + \ldots + |x_{nf} - m_f|\right),$$

$$z(x_{if}) = \frac{x_{if} - m_f}{s_f}.$$

Data Standardization

Ratio-scaled attributes:

- Numeric attributes, but unlike interval-scaled attributes, *their scales are exponential*,
- For example, the total amount of microorganisms that evolve in a time *t* is approximately given by

  $$Ae^{Bt},$$

  - where *A* and *B* are some positive constants.
- Do *log transform*:

  $$\log(x_{if})$$

  - Then treat it as an interval-scaled attribute

K-Medoids Clustering

K-Medoids Clustering

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

K-Medoids:

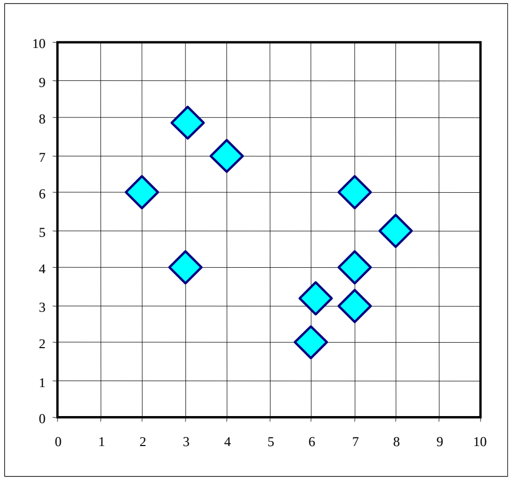
- Instead of taking the mean value of the object in a cluster as a reference point, *medoids can be used*, which is the most centrally located object in a cluster.

K-Medoids Clustering

- Find *representative* objects, called **medoids**, in the clusters

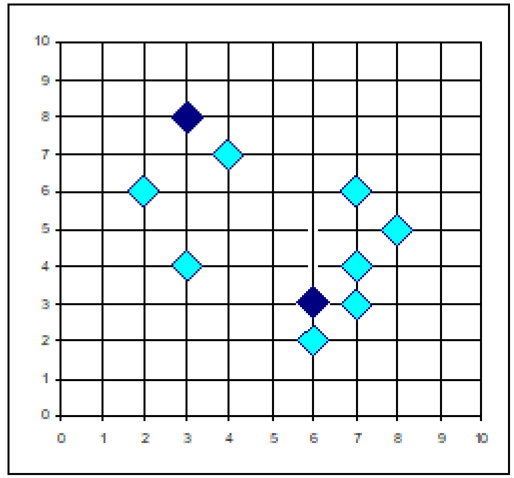
- ***PAM* (Partitioning Around Medoids, 1987)**
  - starts from an initial set of medoids and iteratively replaces one of the medoids by one of the non-medoids if it improves the total distance of the resulting clustering
  - *PAM* works effectively for small data sets, but does not scale well for large data sets

K-Medoids Clustering



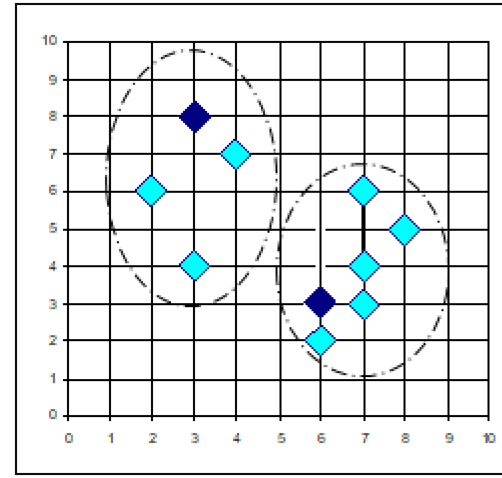
K=2

Arbitrary choose k object as initial medoids

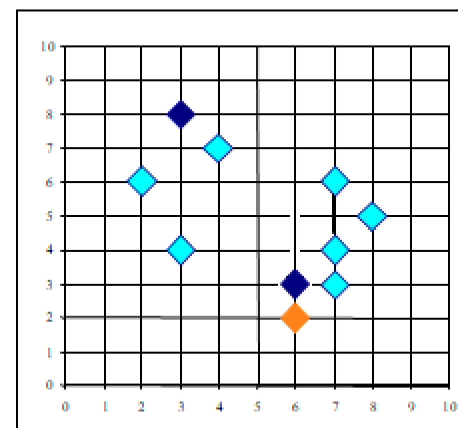


Assign each remaining object to nearest medoids

Total Cost = 20

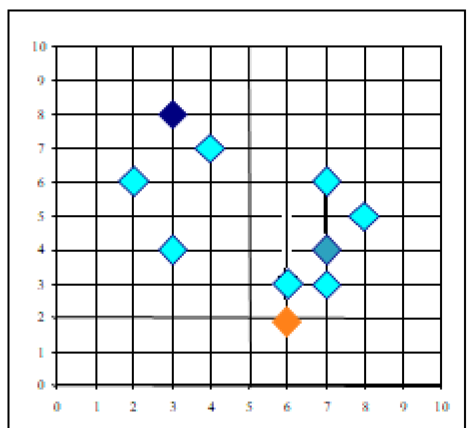


Randomly select a nonmedoid object, O_{ramdom}



Compute total cost of swapping

Total Cost = 26



Swapping O and O_{ramdom}

If quality is improved.

Do loop

Until no change

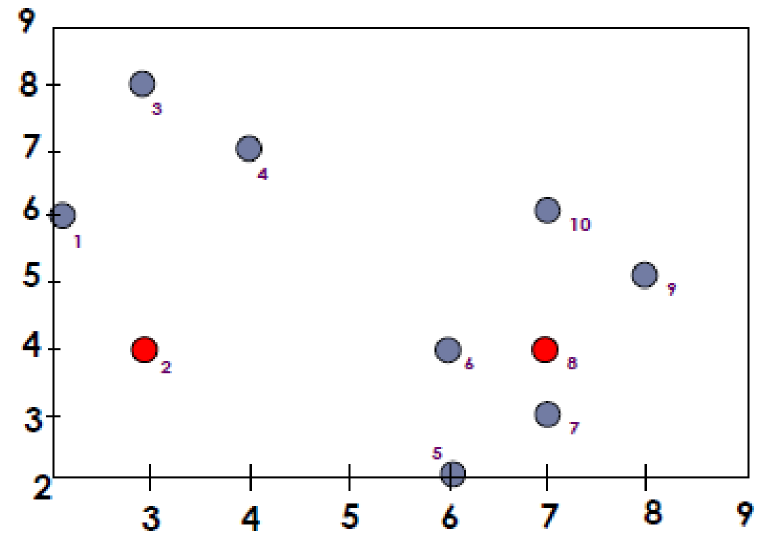
K-Medoids Clustering

- Use real object to represent the cluster
  1. Select ***k*** representative objects arbitrarily
  2. For each pair of non-selected object $\boldsymbol{h}$ and selected object $\boldsymbol{i}$, calculate the **total swapping cost** $\boldsymbol{TC_{ih}}$
  3. For each pair of $\boldsymbol{i}$ and $\boldsymbol{h}$,
     - If $TC_{ih} < 0$, $\boldsymbol{i}$ is replaced by $\boldsymbol{h}$
     - Then assign each non-selected object to the most similar representative object
  4. repeat steps 2-3 until there is no change

K-Medoids Clustering

Data Objects

	A_1	A_2
O_1	2	6
O_2	3	4
O_3	3	8
O_4	4	7
O_5	6	2
O_6	6	4
O_7	7	3
O_8	7	4
O_9	8	5
O_{10}	7	6



Goal: create two clusters

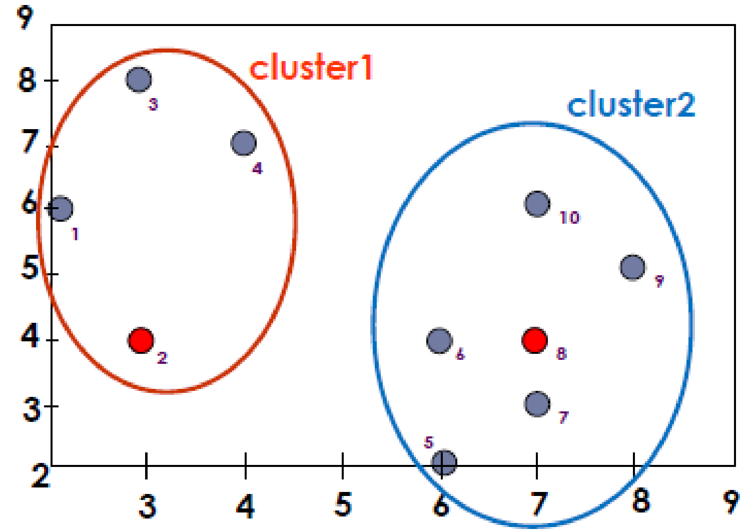
Choose randmly two medoids

$O_2 = (3,4)$
$O_8 = (7,4)$

K-Medoids Clustering

Data Objects

	A_1	A_2
O_1	2	6
O_2	3	4
O_3	3	8
O_4	4	7
O_5	6	2
O_6	6	4
O_7	7	3
O_8	7	4
O_9	8	5
O_{10}	7	6



→Assign each object to the closest representative object

→Using L1 Metric (Manhattan), we form the following clusters

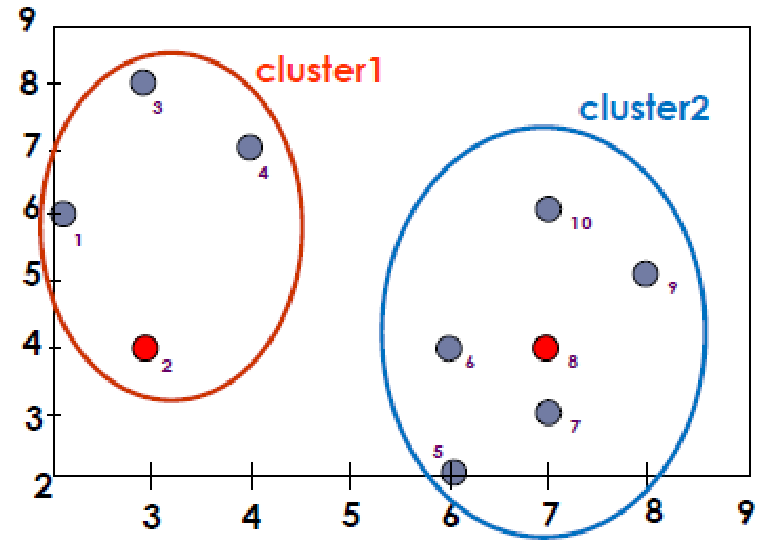
Cluster1 = $\{O_1, O_2, O_3, O_4\}$

Cluster2 = $\{O_5, O_6, O_7, O_8, O_9, O_{10}\}$

K-Medoids Clustering

Data Objects

	A_1	A_2
O_1	2	6
O_2	3	4
O_3	3	8
O_4	4	7
O_5	6	2
O_6	6	4
O_7	7	3
O_8	7	4
O_9	8	5
O_{10}	7	6



→Compute the absolute error criterion **[for the set of Medoids (O2,O8)]**

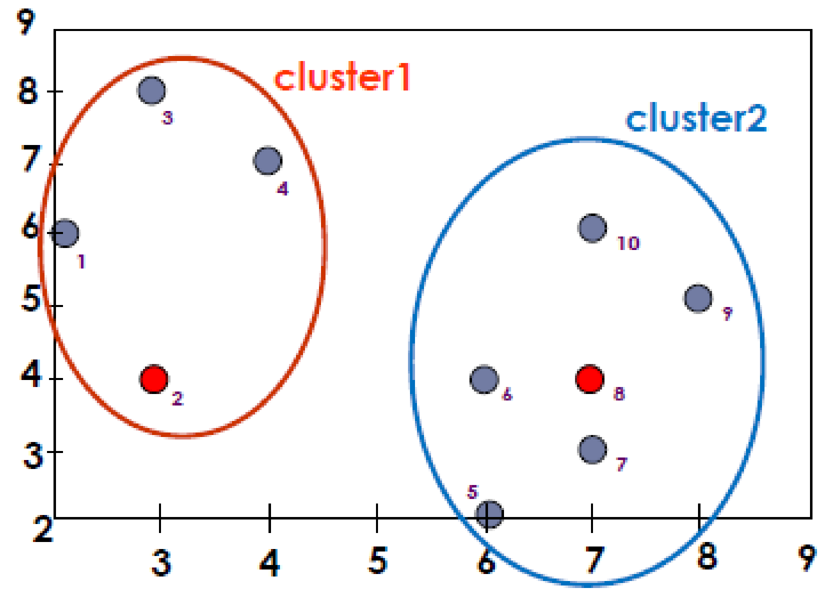
$$E=\sum_{i=1}^{k}\sum_{p\in C_i}|p-o_i|=|o_1-o_2|+|o_3-o_2|+|o_4-o_2|$$

$$+|o_5-o_8|+|o_6-o_8|+|o_7-o_8|+|o_9-o_8|+|o_{10}-o_8|$$

K-Medoids Clustering

Data Objects

	A_1	A_2
O_1	2	6
O_2	3	4
O_3	3	8
O_4	4	7
O_5	6	2
O_6	6	4
O_7	7	3
O_8	7	4
O_9	8	5
O_{10}	7	6



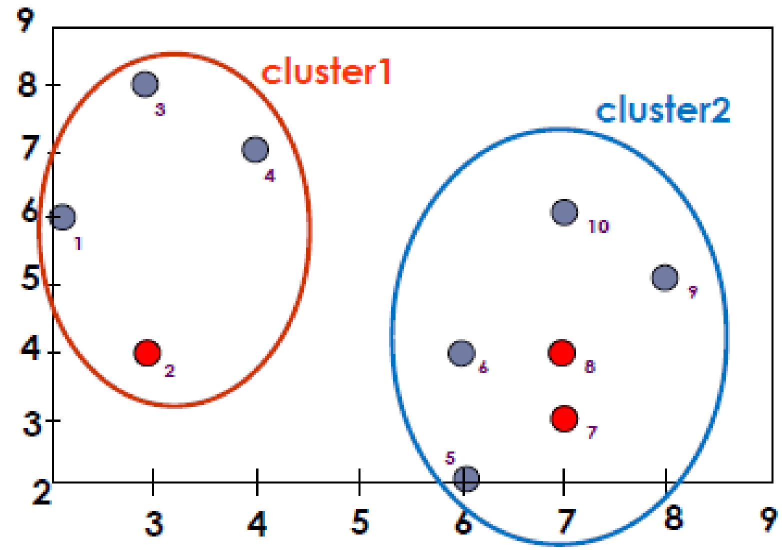
→The absolute error criterion **[for the set of Medoids (O2,O8)]**

$$E = (3+4+4)+(3+1+1+2+2) = 20$$

K-Medoids Clustering

Data Objects

	A_1	A_2
O_1	2	6
O_2	3	4
O_3	3	8
O_4	4	7
O_5	6	2
O_6	6	4
O_7	7	3
O_8	7	4
O_9	8	5
O_{10}	7	6



→Choose a random object O_7

→Swap O8 and O7

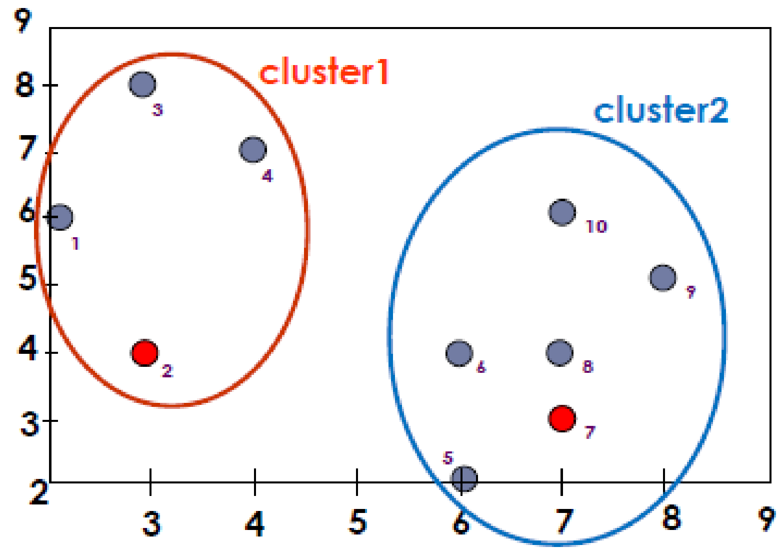
→Compute the absolute error criterion [for the set of Medoids (O2,O7)]

$$E = (3+4+4)+(2+2+1+3+3) = 22$$

K-Medoids Clustering

Data Objects

	A_1	A_2
O_1	2	6
O_2	3	4
O_3	3	8
O_4	4	7
O_5	6	2
O_6	6	4
O_7	7	3
O_8	7	4
O_9	8	5
O_{10}	7	6



→Compute the cost function

Absolute error [for O_2,O_7] – Absolute error [O_2,O_8]

$$S = 22 - 20$$

S> 0 $\Rightarrow$ it is a bad idea to replace O_8 by O_7

K-Medoids Clustering

- *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
- PAM works efficiently for small data sets but **does not scale well** for large data sets.
- $O(k(n-k)^2)$ for each iteration
  - where n is # of data points,
  - k is # of clusters