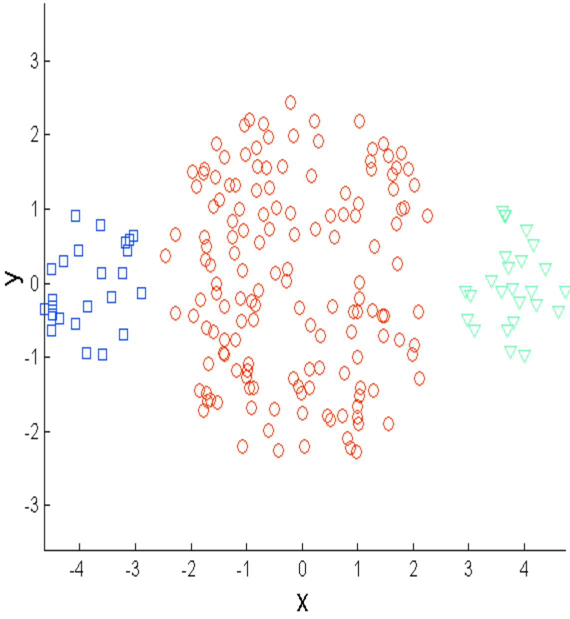
Limitations of K-means

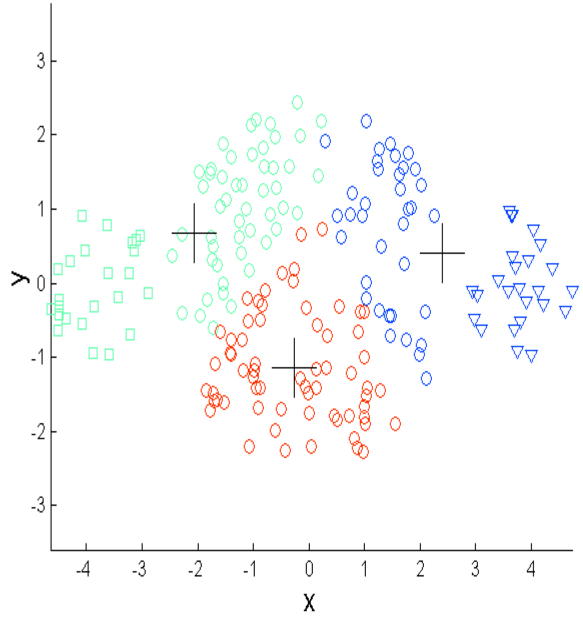
- K-means has problems when clusters are of differing
  - Sizes
  - Densities
  - Non-globular shapes

- K-means has problems when the data contains outliers.

Limitations of K-means: Differing Sizes

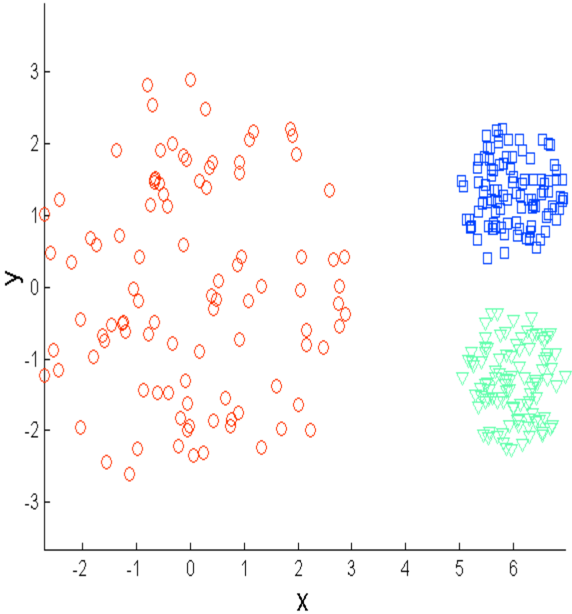


Original Points

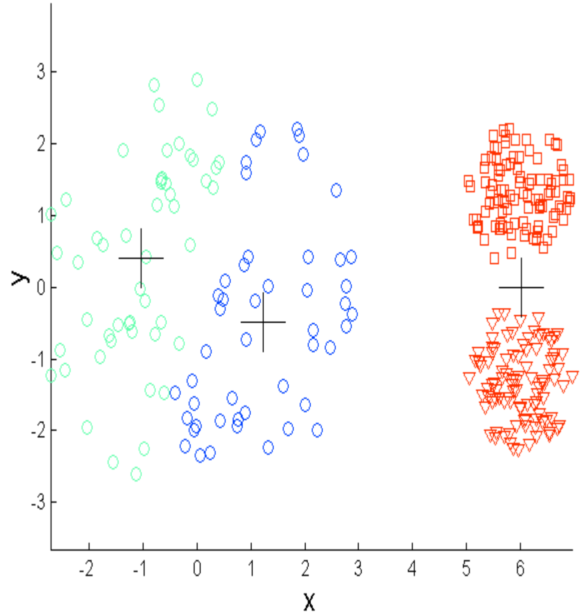


K-means (3 Clusters)

Limitations of K-means: Differing Density

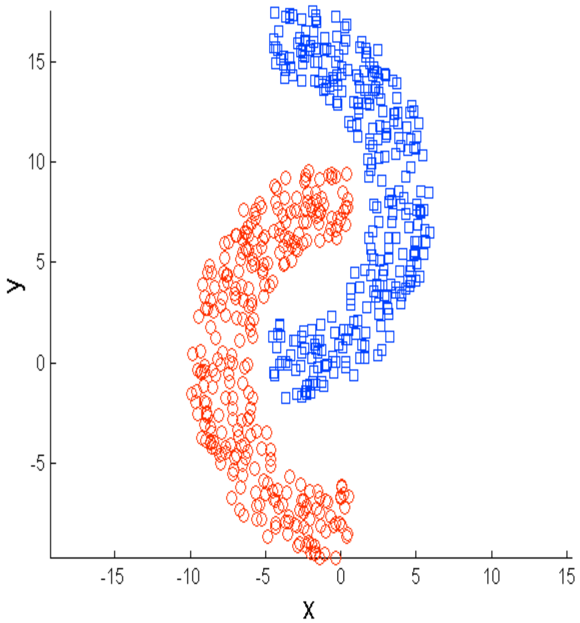


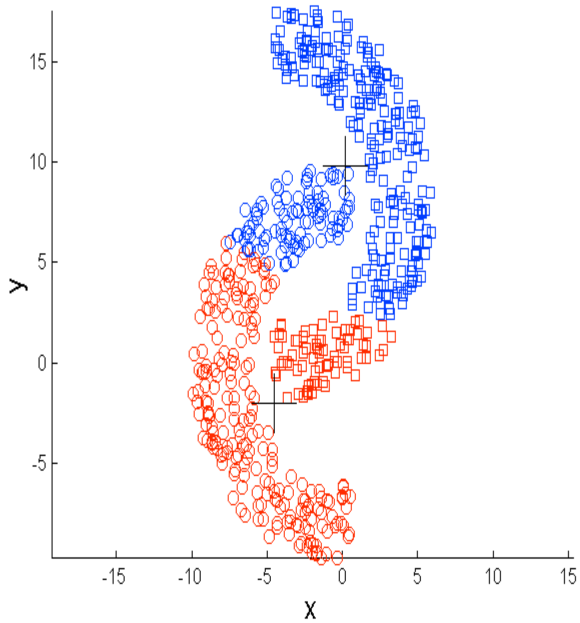
Original
Points



K-means (3 Clusters)

Limitations of K-means: Non-globular Shapes



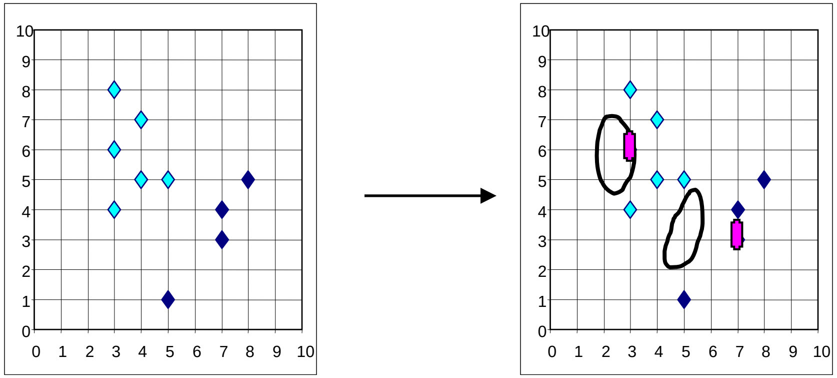


Original Points

K-means (2 Clusters)

Limitations of K-means: Outlier Problem

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



The K-Medoids Clustering Method

- Find *representative* objects, called medoids, in 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.

PAM (Partitioning Around Medoids)

- Use real objects to represent the clusters (called medoids)
  1. Select ***k*** representative objects arbitrarily
  2. For each pair of selected object (***i***) and non-selected object (***h***), calculate the Total swapping Cost (TC_{ih})
  3. For each pair of ***i*** and ***h***,
     1. If $TC_{ih} < 0$, ***i*** is replaced by ***h***
     2. Then assign each non-selected object to the most similar representative object
  4. repeat steps 2-3 until there is no change in the medoids or in TC_{ih}.

Total swapping Cost (TC_{ih})

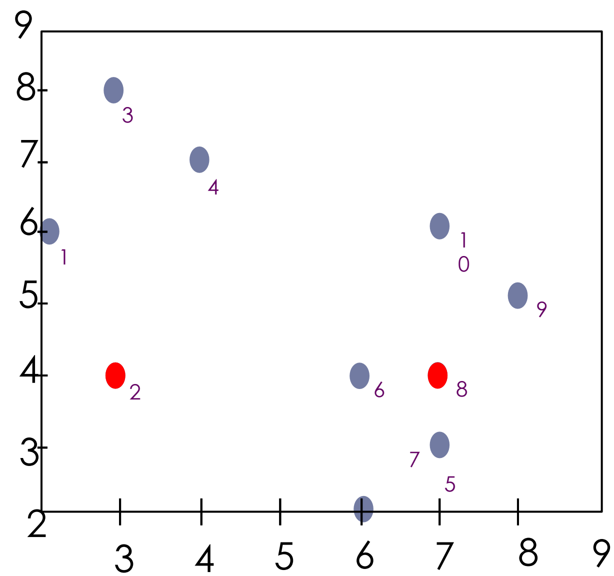
Total swapping cost $TC_{ih} = \sum_j C_{jih}$

- *Where* C_{jih} *is the cost of swapping* i with h *for all non medoid objects* j
- C_{jih} will vary depending upon different cases

PAM or K-Medoids: Example

Data Objects

	A_1	A_2
0_1	2	6
0_2	3	4
0_3	3	8
0_4	4	7
0_5	6	2
0_6	6	4
0_7	7	3
0_8	7	4
0_9	8	5
0_{10}	7	6



Goal: create two clusters

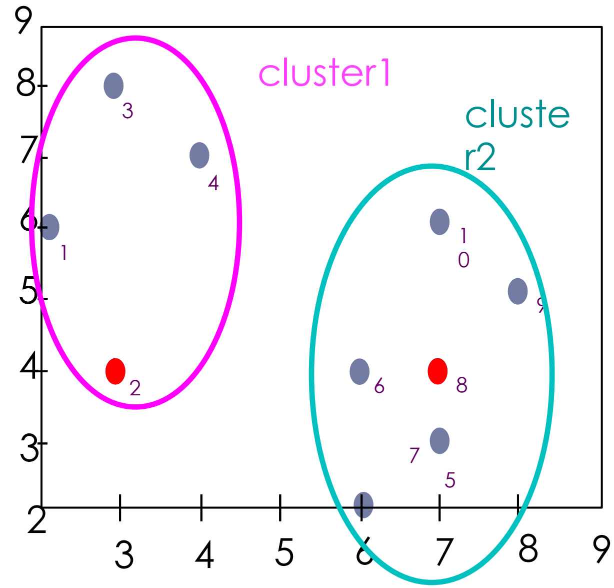
Choose randmly two medoids

$0_8 = (7,4)$ and $0_2 = (3,4)$

PAM or K-Medoids: Example

Data Objects

	A_1	A_2
0_1	2	6
0_2	3	4
0_3	3	8
0_4	4	7
0_5	6	2
0_6	6	4
0_7	7	3
0_8	7	4
0_9	8	5
0_{10}	7	6



Assign each object to the closest representative object

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

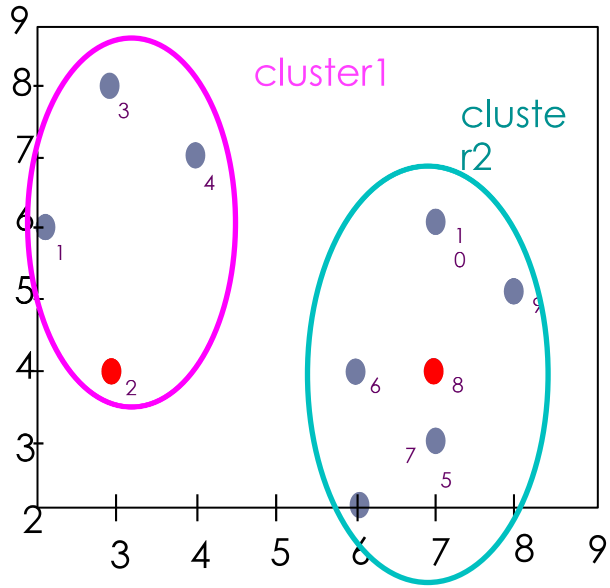
Cluster1 = $\{0_1, 0_2, 0_3, 0_4\}$

Cluster2 = $\{0_5, 0_6, 0_7, 0_8, 0_9, 0_{10}\}$

PAM or K-Medoids: Example

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)]

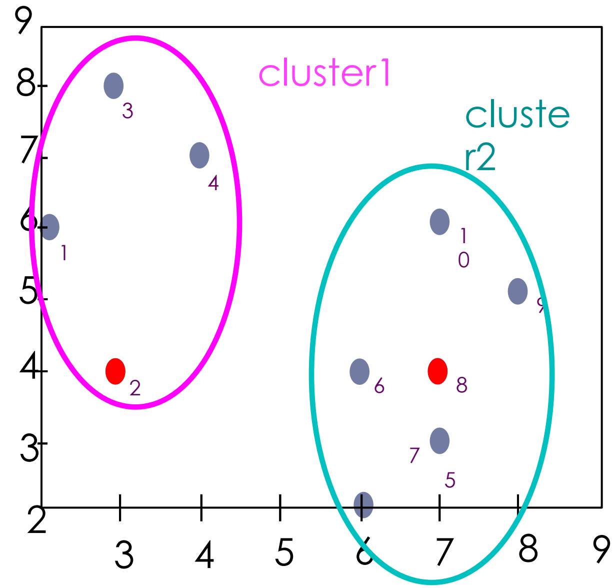
$$E = \sum_{i=1}^{k} \sum_{p \in C_i} |p - O_i| = (|O_1 - O_2| + |O_3 - O_2| + |O_4 - O_2|) +$$

$$(|O_5 - O_8| + |O_6 - O_8| + |O_7 - O_8| + |O_9 - O_8| + |O_{10} - O_8|)$$

PAM or K-Medoids: Example

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



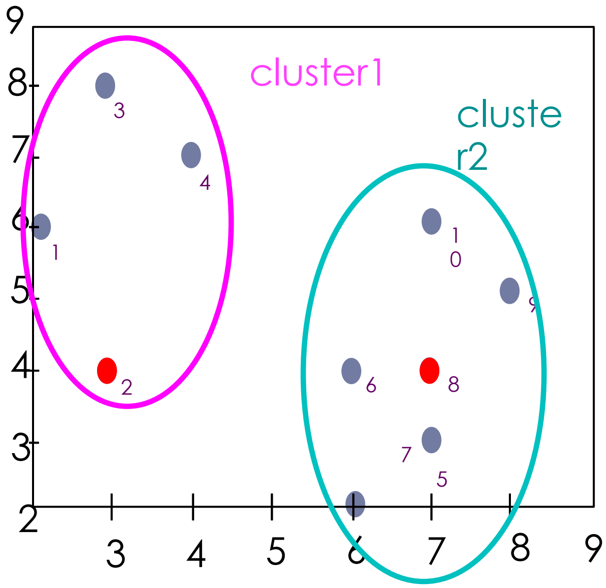
The absolute error criterion [for the set of Medoids (O_2, O_8)]

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

PAM or K-Medoids: Example

Data Objects

	A_1	A_2
0_1	2	6
0_2	3	4
0_3	3	8
0_4	4	7
0_5	6	2
0_6	6	4
0_7	7	3
0_8	7	4
0_9	8	5
0_{10}	7	6



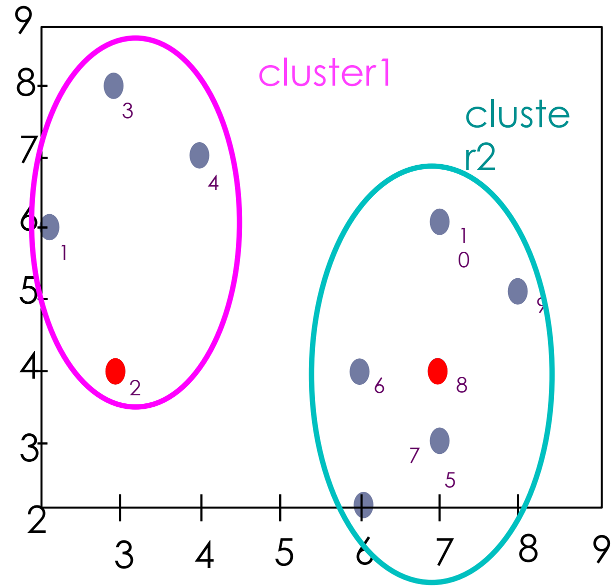
- Choose a random object 0_7
- Swap 0_8 and 0_7
- Compute the absolute error criterion [for the set of Medoids $(0_2, 0_7)$

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

PAM or K-Medoids: Example

Data Objects

	A_1	A_2
0_1	2	6
0_2	3	4
0_3	3	8
0_4	4	7
0_5	6	2
0_6	6	4
0_7	7	3
0_8	7	4
0_9	8	5
0_{10}	7	6



→ Compute the cost function

Absolute error [0_2 ,0_7] - Absolute error [for 0_2 ,0_8]

$$S=22-20$$

S> 0 => It is a bad idea to replace 0_8 by 0_7

What Is the Problem with PAM?

- 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, k is # of clusters.