K-MEANS CLUSTERING

Sourav Karmakar

souravkarmakar29@gmail.com

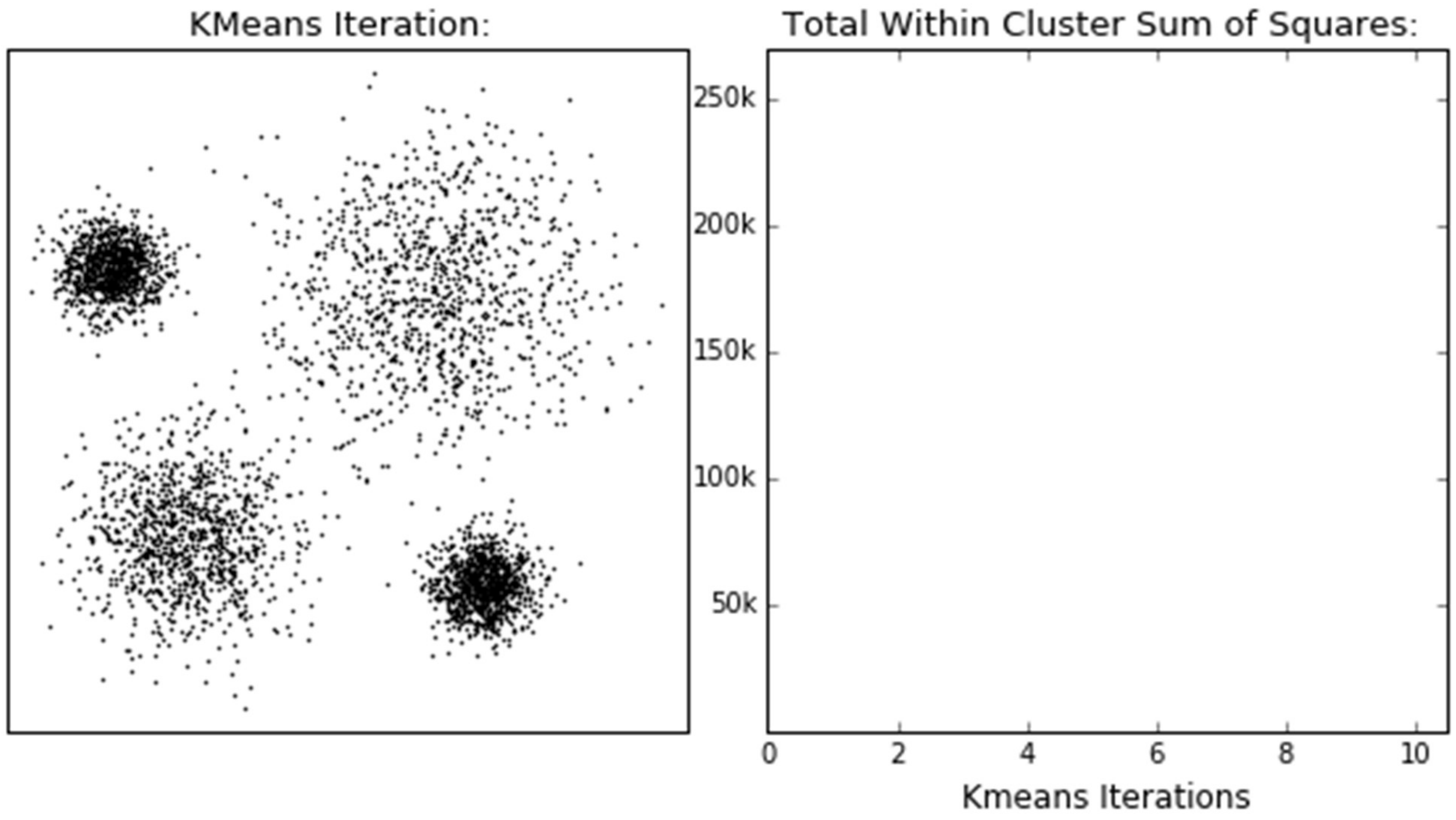
K-MEANS CLUSTERING

- K-Means is a partitional clustering algorithm.
- The K-means algorithm partitions the given data into K clusters.
  - Each cluster has a cluster centre, called centroid.
  - K is user specified.
- One should always scale the data before applying any clustering techniques.
- K-Means algorithm:
  1. Select K points randomly as initial centroids.
  2. ***repeat***
  3. Form K clusters $\{C_1, C_2, \ldots, C_K\}$ by assigning all points to the closest centroid.
  4. Recompute the centroid of each cluster using the formula:

$$\vec{\boldsymbol{\mu}}_{C_i} = \frac{1}{|C_i|} \sum_{\vec{\boldsymbol{x}} \in C_i} \vec{\boldsymbol{x}} \quad , where\ |C_i|\ denotes\ number\ of\ points\ in\ Cluster - i$$

  5. ***until*** the centroids don't change or no reassignment of data points in different clusters. (convergence)

K-MEANS CLUSTERING



K-MEANS CLUSTERING

- Total *Within Cluster Sum of Squares (WCSS)* is obtained by following formula:

$$WCSS = \sum_{j=1}^{K} \sum_{\vec{x} \in C_j} dist\left(\vec{\boldsymbol{x}}, \vec{\boldsymbol{\mu}}_j\right)^2$$

Where, $\vec{\boldsymbol{\mu}}_j$ is the centroid of the cluster C_j and there are K such clusters.
Here, $dist(\dots)$ denotes the distance function of user's choice. (usually Euclidean distance)

Some remarks about K-Means:

1. As initial centroids are often chosen randomly the cluster produced may vary from one run to another.
2. K-means will converge for more common similarity / dissimilarity measures.
3. Most of the convergence happens in the first few iterations.
4. Complexity of the algorithm is: $O\ (n \times K \times d \times I)$
   Where, n = number of data points
   K = no. of clusters
   d = dimension of the dataset / number of features
   I = number of iterations

HOW TO CHOOSE 'K': ELBOW METHOD

- The WCSS is a function of 'K'. With Euclidean distance, the WCSS function is following:

$$WCSS\ (K) = \sum_{j=1}^{K} \sum_{\vec{x} \in C_j} \left|\left|\vec{x} - \vec{\mu}_j\right|\right|^2 \ ;\ where\ \vec{\mu}_j = \frac{1}{|C_j|} \sum_{\vec{x} \in C_j} \vec{x}$$

- For $K = 1$:

$$WCSS(1) = \sum_{i=1}^{n} ||\vec{x}_i - \vec{\mu}||^2 \ ;\ where\ \vec{\mu} = \frac{1}{n} \sum_{i=1}^{n} \vec{x}_i \ \ is\ the\ global\ mean$$

- For $K = n$: (each point is its own cluster)

$$WCSS(n) = 0$$

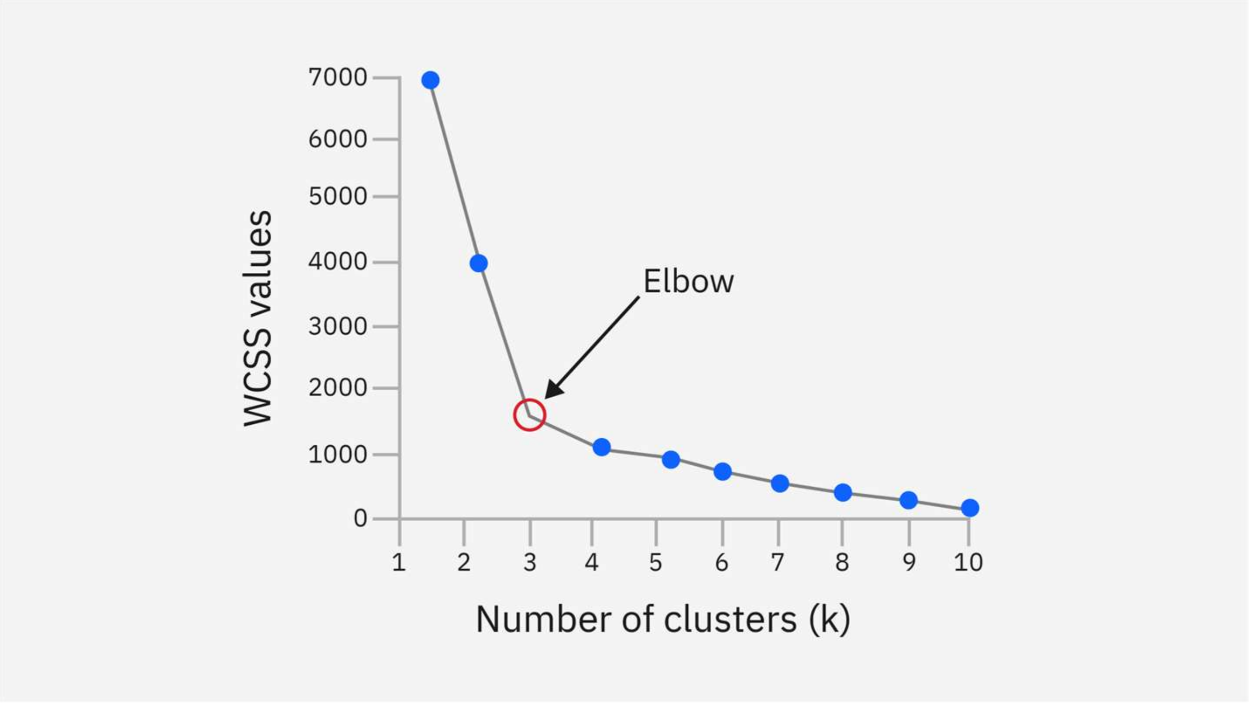
- As K increases, $WCSS\ (K)$ decreases (since more cluster reduce distances). $WCSS(K)$ is a monotonically decreasing function of K.

- But adding clusters beyond a point doesn't give much improvement.

- The "elbow" is the value of K, where the marginal gain: $\Delta_k WCSS = WCSS(K-1) - WCSS(K)$ drops sharply; but after that there is no sharp decrease in WCSS.

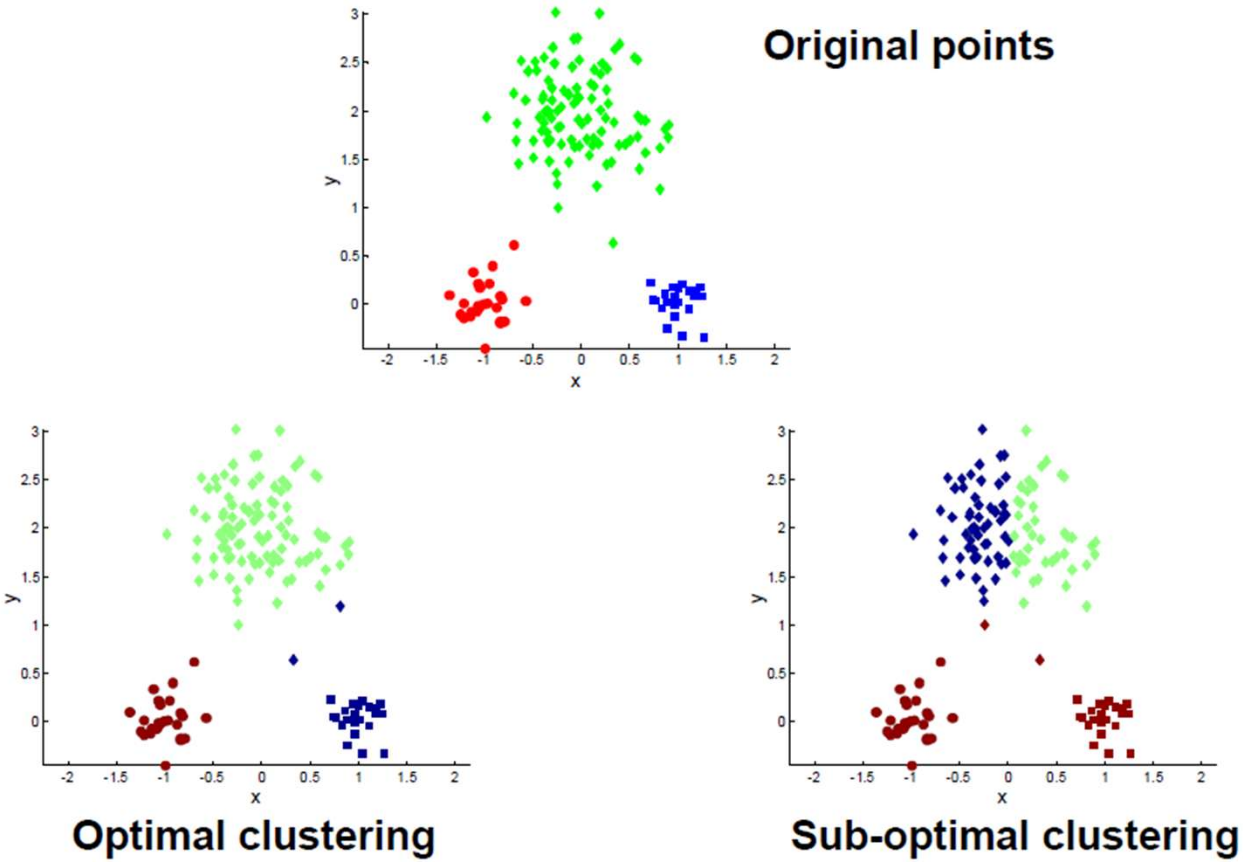
HOW TO CHOOSE ‘K’: ELBOW METHOD

- See the following example:
  - Decrease of WCSS from K=1 to 2: 3000
  - Decrease of WCSS from K=2 to 3: 2500
  - Decrease of WCSS from K=3 to 4: 500 ; and the change is lesser with increasing value of K
- So, Elbow point is K=3. After that the marginal gain diminishes.



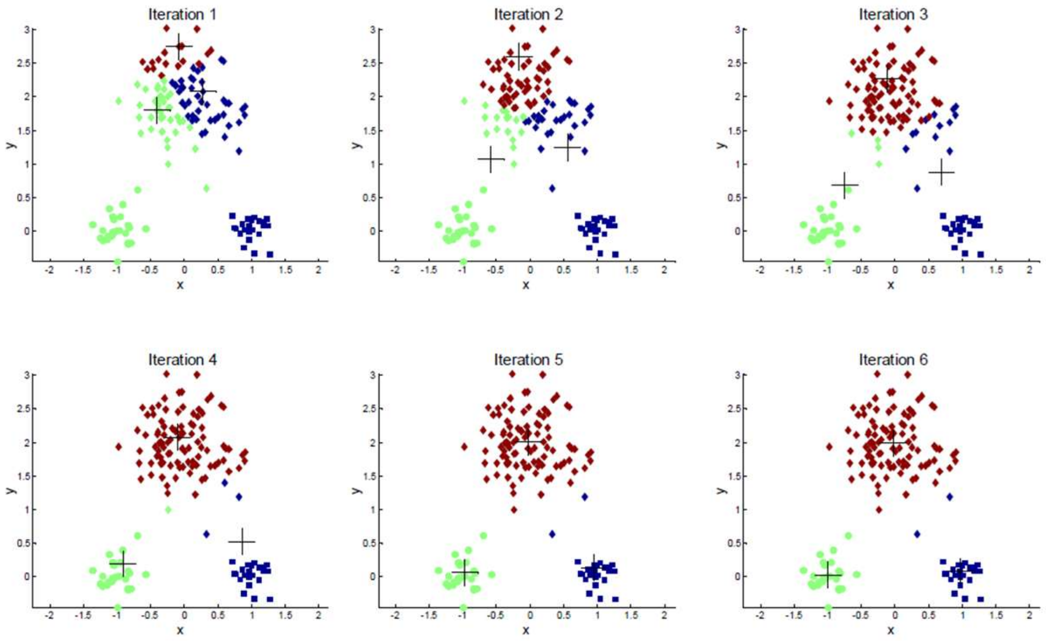
K-MEANS CLUSTERING - LIMITATIONS

- Different runs of the K-means algorithm on the same dataset can produce very different results. This is because random selection of initial centroids.

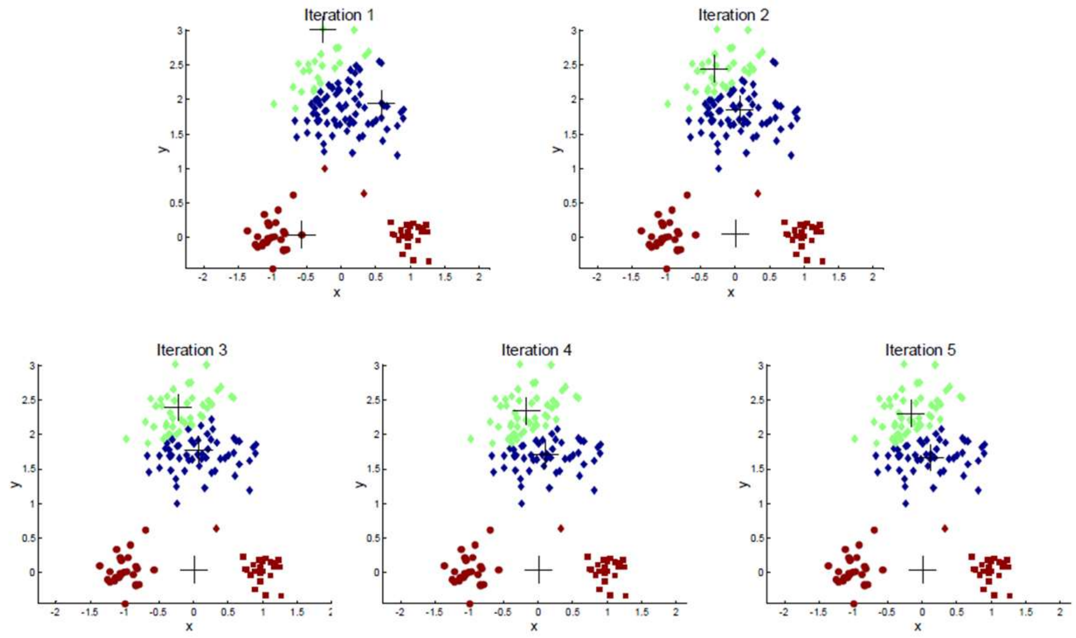


K-MEANS CLUSTERING - LIMITATIONS

RUN-1



RUN-2



Notice that choosing different set of initial centroids have produced completely different clustering on the same dataset.

K-MEANS CLUSTERING - LIMITATIONS

- **Solution to the initial centroid problem**

1. Pre-process the data.
   - Normalize / Standardize the data.
   - Eliminate outliers if possible.

2. Sample the dataset and use *Hierarchical Clustering* (To be discussed in another lecture) to determine the initial centroids.

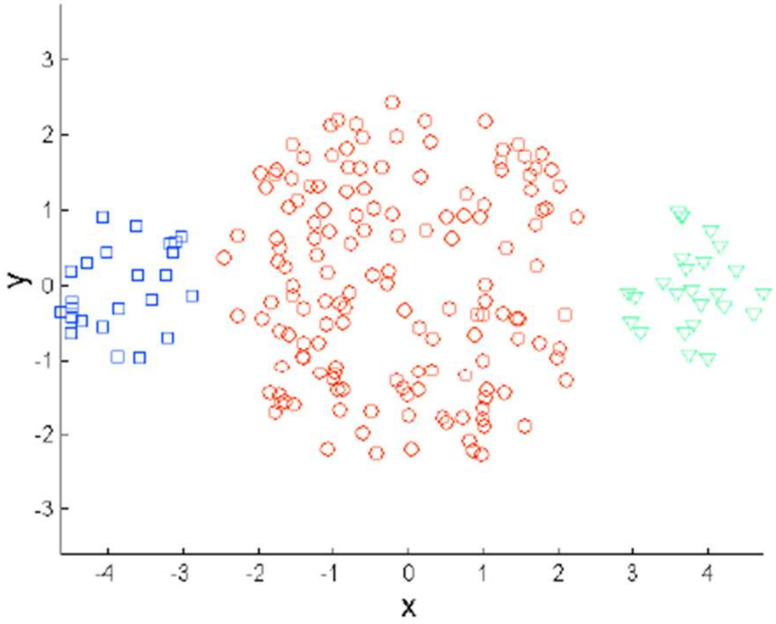
3. Select more than K initial centroids and from these select K most widely separated centroids after clustering.

4. Multiple runs and select the one which gives minimum WCSS value.

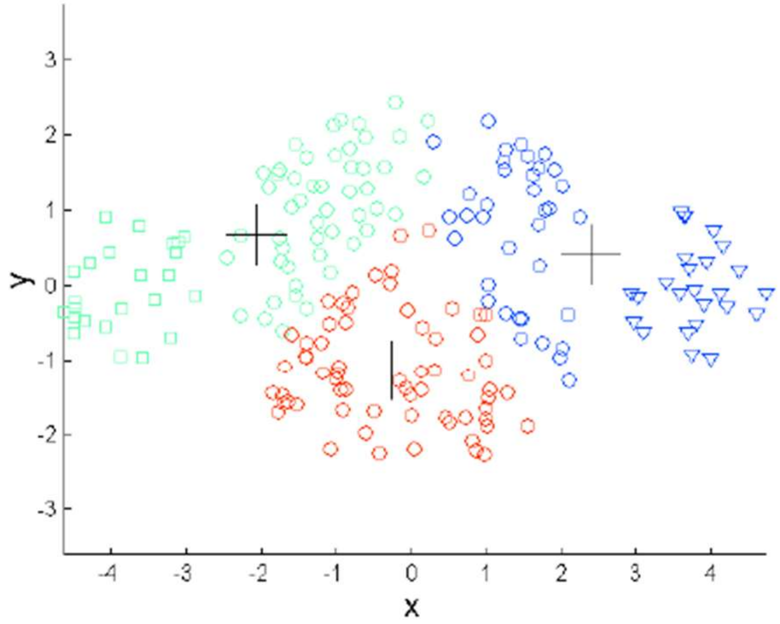
5. Post-process the data.
   - Eliminate small clusters that may represent outliers or noises.
   - Split ‘loose’ clusters, i.e., clusters with relatively high Sum of Square Distances (SSD).
   - Merge clusters that are ‘close’ and that have relatively low SSD.

K-MEANS CLUSTERING - LIMITATIONS

- K-Means faces problem when the clusters are of **different sizes**.



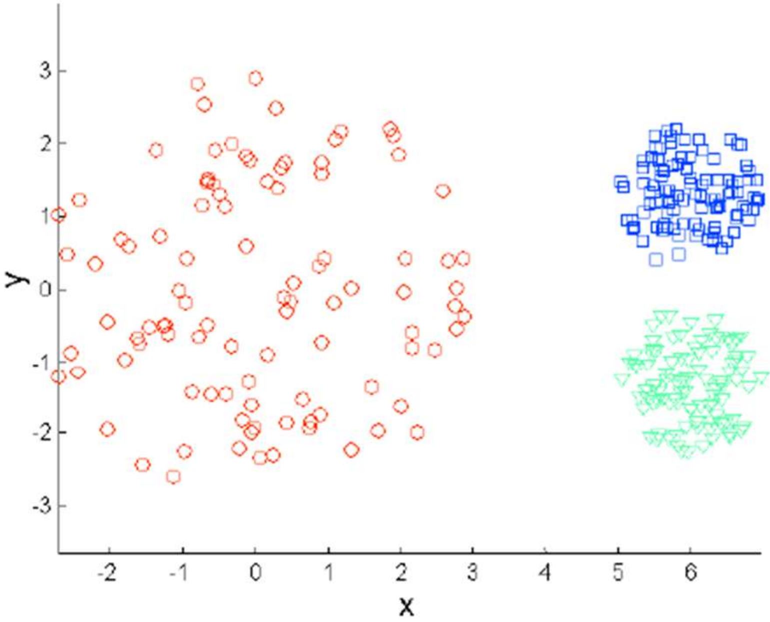
Original Points



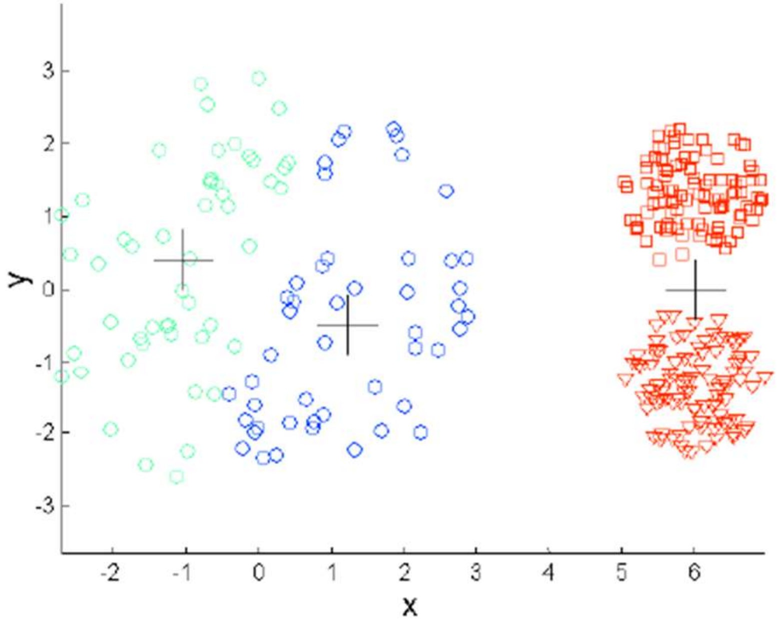
K-means (3 Clusters)

K-MEANS CLUSTERING - LIMITATIONS

- K-Means doesn’t work well when the clusters are of **different densities**.



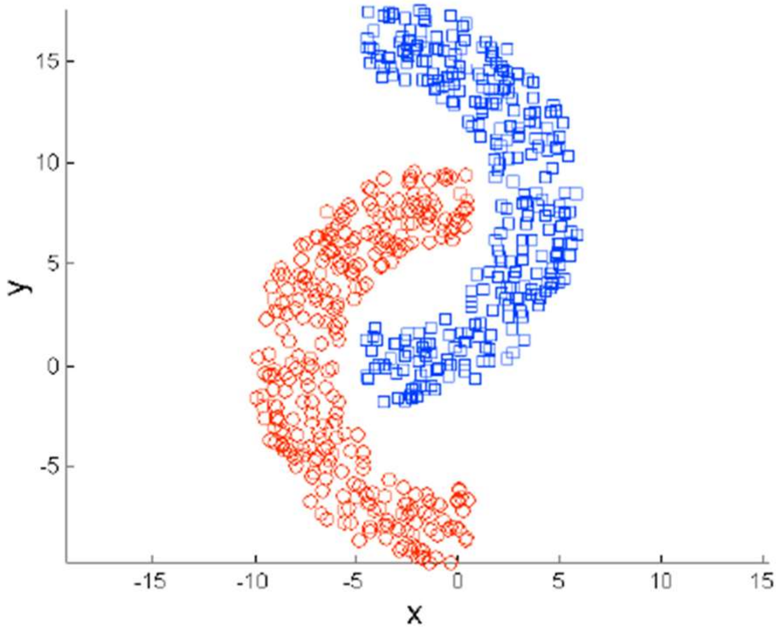
Original Points



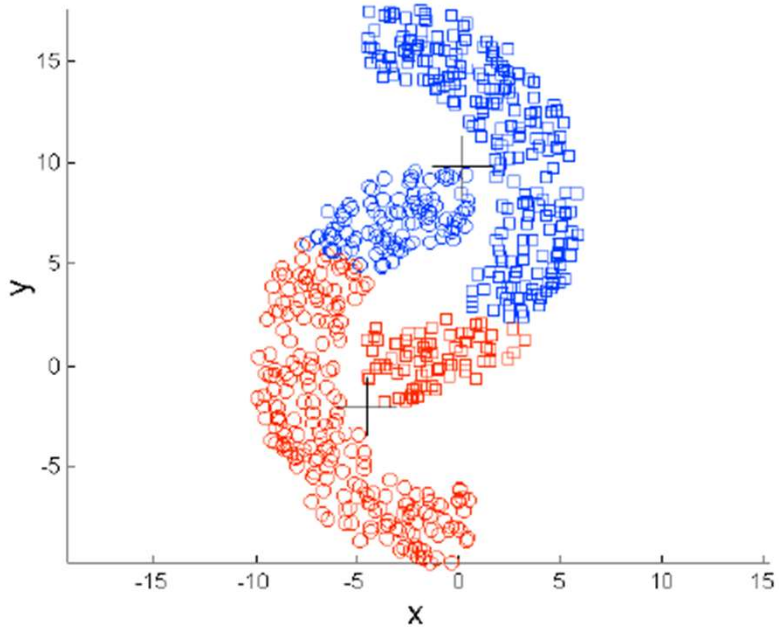
K-means (3 Clusters)

K-MEANS CLUSTERING - LIMITATIONS

- K-Means works well when the clusters are of spherical shape. But struggles when the clusters are of **non-spherical shape**.



Original Points



K-means (2 Clusters)

Thank You