Partitioning-Based Clustering Algorithms

Dr. JASMEET SINGH
ASSISTANT PROFESSOR
CSED, TIET

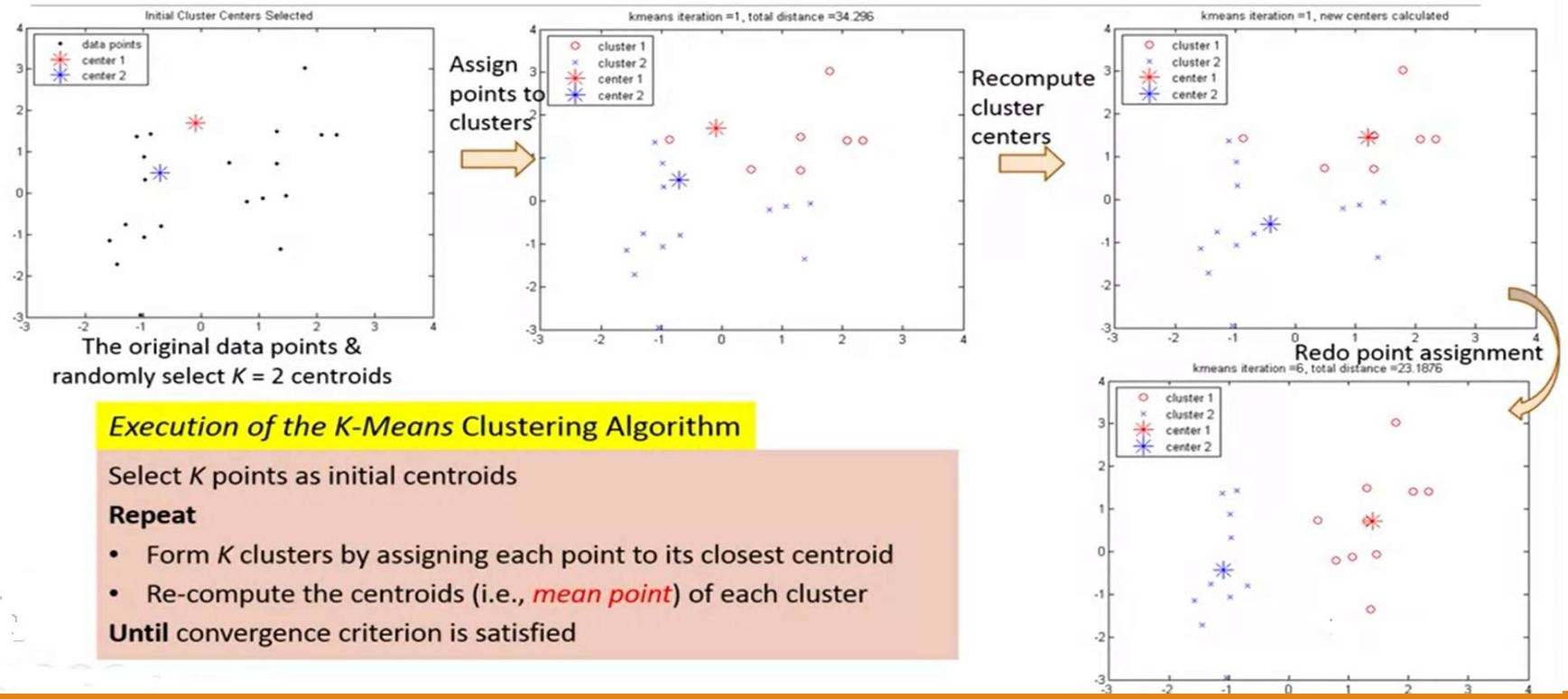
Partitioning-Based Algorithms- Introduction

- Discovers the grouping in the data by optimizing a specific objective function and iteratively improving the quality of clusters.
- These algorithms partitions a dataset D into K clusters so that the objective function is optimized.
- In order to find global optimal it needs to exhaustively enumerate all the partitions.
- Realistically, we use following heuristic methods (greedy methods) to find the clusters:
  - K-Means
  - K-Medians
  - K-Medoids

K-Means Clustering-Introduction

- It is the most popular and widely clustering method.
- It has been proposed by Macqueen in 1967.
- In K-means clustering algorithm, each cluster is represented by the center of the cluster.
- Given K, the number of clusters, the K-means clustering algorithm is outlined as follows:
  - Select K-points as initial clusters.
  - Repeat until convergence or for fixed number of iterations
    - Assignment Step: Form K-clusters by assigning each point to its closest centroid.
    - Update: Recompute the centroids (i.e., mean point) of each cluster.
- Different kinds of measures such as Manhattan distance (L1 Norm), Euclidean Distance (L2 Norm), cosine distance are used to find the distance of each point from the centroids.

K-Means Clustering- Introduction



Execution of the K-Means Clustering Algorithm

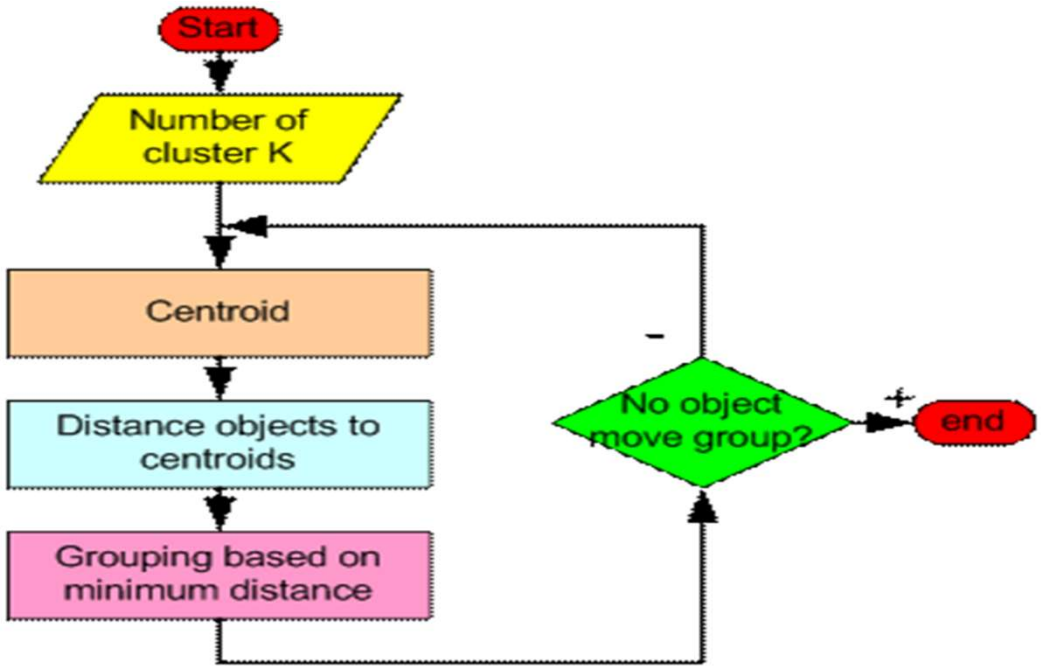
Select *K* points as initial centroids

Repeat

- Form *K* clusters by assigning each point to its closest centroid
- Re-compute the centroids (i.e., *mean point*) of each cluster

Until convergence criterion is satisfied

K-Means- Introduction



K-Means Algorithm

Algorithm 2.1 k-means clustering algorithm.

Input: Data $\mathcal{X} = \{\mathbf{x}_1, ..., \mathbf{x}_n\}$, the order k, MAX number of allowed iterations
Output: A partition $\mathcal{P} = \{\mathcal{C}_1, ..., \mathcal{C}_K\}$

$t = 0, \mathcal{P} = \emptyset$
Randomly initialize $\boldsymbol{\mu}_i, i = 1, ..., K$
loop
    $t+ = 1$
    Assignment Step: assign each sample $\mathbf{x}_j$ to the cluster with the nearest representative
    $\mathcal{C}_i^{(t)} = \{\mathbf{x}_j : d(\mathbf{x}_j, \boldsymbol{\mu}_i) \leq d(\mathbf{x}_j, \boldsymbol{\mu}_h) \text{ for all } h = 1, \ldots, K\}$
    Update Step: update the representatives
    $\boldsymbol{\mu}_i^{(t+1)} = \frac{1}{|\mathcal{C}_i^{(t)}|} \sum_{\mathbf{x}_j \in \mathcal{C}_i} \mathbf{x}_j$
    Update the partition with the modified clusters:
    $\mathcal{P}^t = \{\mathcal{C}_1^{(t)}, ..., \mathcal{C}_K^{(t)}\}$
    if $t \geq$ MAX OR $\mathcal{P}^t = \mathcal{P}^{t-1}$ **then**
        return $\mathcal{P}^t$
    end if
end loop

K-Means- Numerical Example

Suppose we have four types of medicines with two features: weight index, and pH. Group these medicines into 2 groups based on their features using K-Means algorithm.

Medicine	Attribute 1 (X): Weight Index	Attribute II (Y):pH
A	1	1
B	2	1
C	4	3
D	5	4

Example –Solution

K=2, Randomly initialize 2 centers. Let the centers be c1=(1,1) and c2=(2,1)

Iteration-1

Assignment Step: Form K-clusters by assigning each point to its closest centroid.

$$D^0 = \begin{bmatrix} 0 & 1 & 3.61 & 5 \\ 1 & 0 & 2.83 & 4.24 \end{bmatrix} \quad \begin{matrix} c_1 = (1,1) & group-1 \\ c_2 = (2,1) & group-2 \end{matrix}$$

$$G^0 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 \end{bmatrix} \quad \begin{matrix} group-1 \\ group-2 \end{matrix}$$

$$\begin{matrix} A & B & C & D \end{matrix}$$

Update: Recompute the centroids (i.e., mean point) of each cluster.

c1=(1,1) and c2=$\left(\frac{2+4+5}{3}, \frac{1+3+4}{3}\right) = \left(\frac{11}{3}, \frac{8}{3}\right)$

Example –Solution (Contd….)

Iteration-2

Assignment Step: Form K-clusters by assigning each point to its closest centroid.

$$\mathbf{D}^1 = \begin{bmatrix} 0 & 1 & 3.61 & 5 \\ 3.14 & 2.36 & 0.47 & 1.89 \end{bmatrix} \quad \begin{matrix} \mathbf{c}_1 = (1,1) & group-1 \\ \mathbf{c}_2 = (\frac{11}{5}, \frac{8}{5}) & group-2 \end{matrix}$$

$$\mathbf{G}^1 = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix} \quad \begin{matrix} group-1 \\ group-2 \end{matrix}$$
$$A \quad B \quad C \quad D$$

Update: Recompute the centroids (i.e., mean point) of each cluster.

c1= $\left(\frac{1+2}{2}, \frac{1+1}{2}\right) = (\frac{3}{2}, 1)$ and c2=$\left(\frac{4+5}{2}, \frac{3+4}{2}\right) = (\frac{9}{2}, \frac{7}{2})$

Example –Solution (Contd….)

Iteration-3

Assignment Step: Form K-clusters by assigning each point to its closest centroid.

$$\mathbf{D}^2 = \begin{bmatrix} 0.5 & 0.5 & 3.20 & 4.61 \\ 4.30 & 3.54 & 0.71 & 0.71 \end{bmatrix} \begin{matrix} \mathbf{c}_1 = (1\frac{1}{2}, 1) \quad group-1 \\ \mathbf{c}_2 = (4\frac{1}{2}, 3\frac{1}{2}) \quad group-2 \end{matrix}$$

$$\mathbf{G}^2 = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix} \begin{matrix} group-1 \\ group-2 \end{matrix}$$
$$A \quad B \quad C \quad D$$

Update: Recompute the centroids (i.e., mean point) of each cluster.

c1= $\left(\frac{1+2}{2}, \frac{1+1}{2}\right) = (\frac{3}{2}, 1)$ and c2=$\left(\frac{4+5}{2}, \frac{3+4}{2}\right) = (\frac{9}{2}, \frac{7}{2})$

Since, there is no change in centroids and clusters. Hence, the algorithm will stop. So, the final clusters are (Medicine A, Medicine B) and (Medicine C, Medicine D)

K-Mean Objective Function

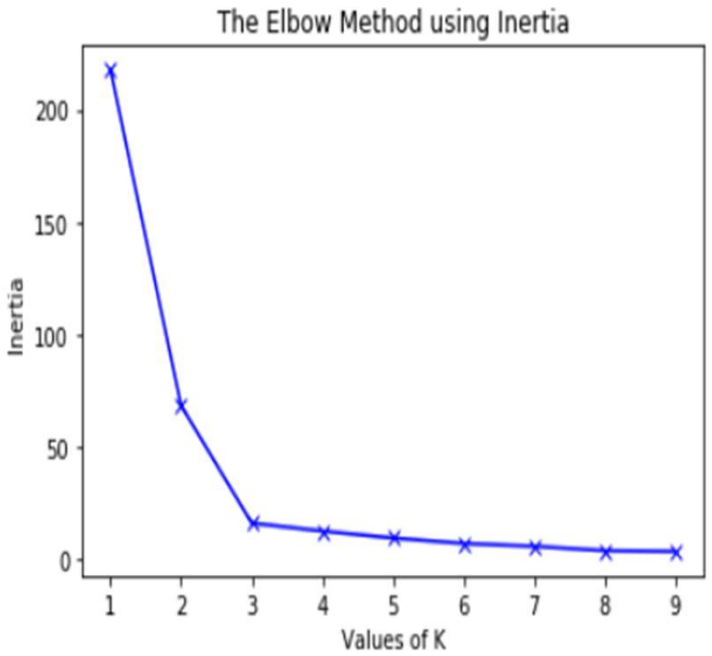
- K-Means algorithm partitions a dataset D of n objects into a set of K clusters so that the objective function is maximized.
- Particularly, it uses sum of squared error or deviations of each sample from the center of the cluster to which it belongs.
- The sum squared error is given by:

$$SSE(C) = \sum_{k=1}^{K} \sum_{\forall x_i \epsilon c_k} |x_i - c_k|^2$$

- The above mean squared error is also called inertia and the mean of the above function is called distortion.
- The SSE strictly decreases after we recompute new centers in the k-means algorithm because the new center of a cluster comes from the average of all data points in this cluster, which actually minimizes the SSE.

How to choose K?

- There are variety of ways to find the optimal numbers of clusters (i.e. K)
- The most popular method is the ***elbow method.***
- In this method, we consider number of distinct values of K, and for each value of K, the sum squared deviations of samples from the cluster center is computed.
- To determine the optimal number of clusters, we have to select the value of k at the "elbow" i.e. the point after which the distortion/inertia start decreasing in a linear fashion.
- For instance, in the figure, we conclude that the optimal number of clusters for the data is **3**.



How to choose K? (Contd.....)

- We can also find the optimal number of clusters using the clustering evaluation metrics.
- We can choose different values of number of clusters (K), and for each value of K, silhouette coefficient is computed. The value for which Silhouette coefficient is maximum is chosen.
- In case the ground truth is available, we can also consider external evaluation metrics such as accuracy, Rand Index, Adjusted Rand Index, Purity, etc.

K-Means: Key Points

- Efficiency: O(tKn): where n are number of objects, K are number of clusters, and t is the number of iterations.
  - Normally K,t << n; thus an efficient method.
- K-means clustering often ***terminates at a local optimal***.
  - Initialization can be important to find high quality clusters.
- Sensitive to ***noisy data and outliers***.
  - Variations: K-medians, K-medoids, etc.
- K-means is applicable only to objects in a continuous n-dimensional space.
  - Using the K-modes for categorical data.
- Not suitable to discover clusters with non-convex shapes.
  - Use density-based methods, kernel K-means, etc.

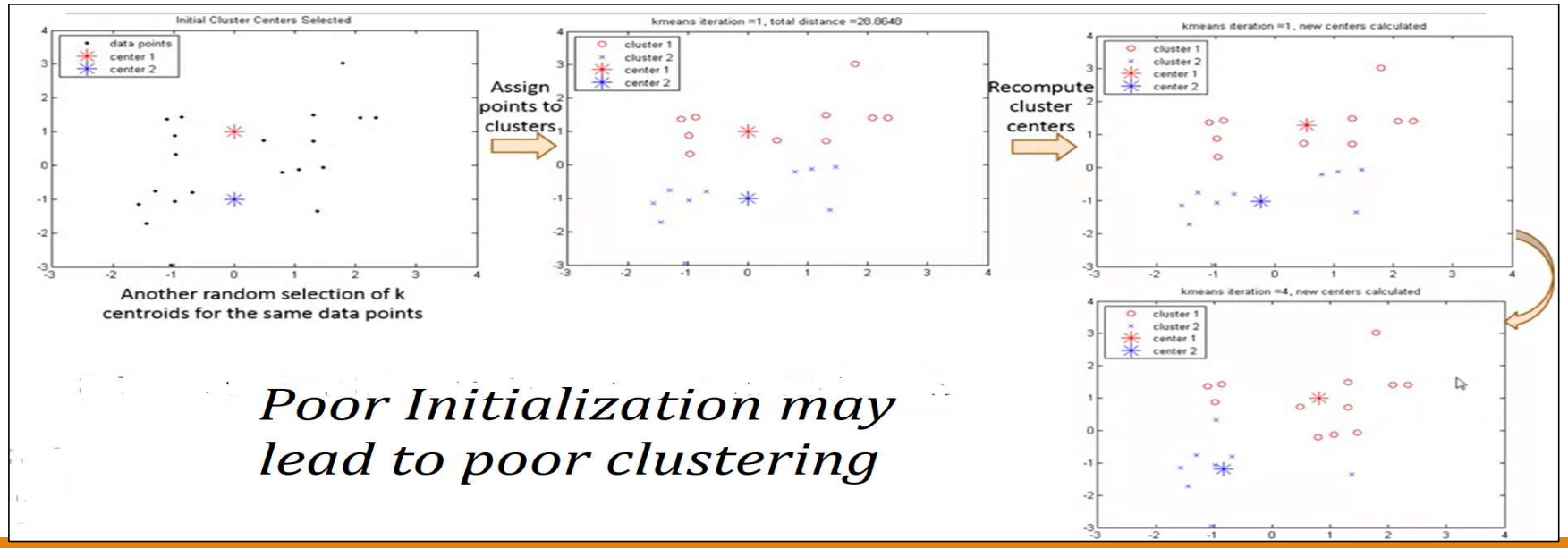
Variations of K-Means

- There are many variations of the K-means method, varying in different aspects

1. Choosing better initial cluster centroids.
   - K-means ++, Intelligent K-means, Genetic K-means
2. Choosing different representative prototypes for the clusters.
   - K-medians, K-medoids, K-modes
3. Applying feature transformation techniques
   - Weighted K-means, Kernel K-Means

Initialization of K-Means

- in K-means, the initial centroids are randomly initialized. Some of these initialization may not lead to global optimal (i.e. minimum sum square error) but may stuck into local minimum.



Initialization of K-Means (Contd….)

- There are two particular solutions to the random initialization problem.

1. Original Proposal (MacQueen'67): select k seeds randomly
   - Need to run the algorithm multiple times using different seeds.
2. Use advanced versions of K-means for better initialization of k-seeds
   - K-Means++
   - Genetic K-Means
   - Intelligent K-Means

K-Means++

- K-Means ++ algorithm has been proposed by Arthur and Vassilvitskii (2007).
- It is different from the K-means algorithm in the way it update the centroids (i.e. the update centroid step).
- The K-means++ algorithm is as follows:
  1. Randomly select the first centroid from the data points.
  2. For each data point compute its distance from the nearest, previously chosen centroid.
  3. Select the next centroid from the data points such that the probability of choosing a point as centroid is directly proportional to its distance from the nearest, previously chosen centroid. (i.e. the point having maximum distance from the nearest centroid is most likely to be selected next as a centroid)
  4. Repeat steps 2 and 3 until k centroids have been sampled
- It solves the problem of random initialization as the new centroid is having maximum probability proportional to distance from all the points.

K-Medoid Clustering

- K-Medoid clustering is also called PAM (Partitioning around Medoids).
- PAM algorithm has been proposed by Kaufmann and Rousseeuw in 1987
- K-Means algorithm is sensitive to outliers- since an object with an extreme value may substantially distort the distribution of data.
- K-Medoid algorithm instead of taking the mean value of objects in a cluster, uses medoids, which is the most centrally located object in the cluster.
- PAM algorithm works as follows:
  1. Starts from an initial set of medoids (randomly chosen).
  2. Repeat until convergence or for fixed number of iterations
     - Assignment: Assign the data points to the nearest medoid.
     - Update: Replace each medoid in each cluster by one of the non-medoid of the same cluster if it improve the sum square error of the resulting cluster.

K-Medoid: Key Points

- PAM algorithm works effectively for small data sets but does not scale well for large data sets (due to computational complexity).
- Computational complexity: PAM: $O(K(n-K)^2)$ – quite expensive (where n are number of objects; and K are number of clusters)
- Efficiency improvements on PAM
  - CLARA – PAM on random samples – $O(Ks^2+K(n-K))$; s is the sample size.
  - CLARANS - PAM with randomized resampling, ensure efficiency and quality

K-Median

- Medians are less sensitive to outliers than means.
  - For instance, median salary is less affected as compared to mean salary of a large firm with few top executives.
- In K-Medians, instead of taking the mean value of the object in a cluster as a reference point, medians are used.
- The objective function that is minimized by the K-Median algorithm is as follows:

$$SSE(C) = \sum_{k=1}^{K} \sum_{\forall x_i \epsilon c_k} |x_i - median_k|^2$$

- K-Median function uses L1 normalization (Manhattan distance) to assign the data points to the nearest median.

K-Median (Contd……)

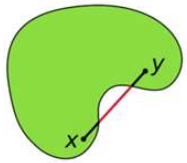
- K-Median algorithm works in the following steps:
  1. Select K points as the initial representative objects (i.e. initial K Medians).
  2. Repeat until convergence or fixed number if iterations:
     - Assignment: Assign the data points to the nearest median.
     - Update: Recompute the median of each cluster using the median of the datapoints in the individual clusters.

K-Mode

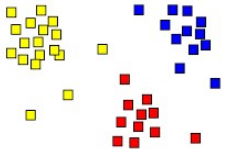
- K-Means cannot handle non-numerical data (categorical data).
  - Mapping categorical values to numerical values does not generate quality clusters for high dimensional data.
- K-modes is an extension to K-means by replacing means of clusters with modes.
- K-Median algorithm works in the following steps:
  1. Select K points as the initial representative objects (i.e. initial K Mode).
  2. Repeat until convergence or fixed number if iterations:
     - Assignment: Assign the data points to the nearest representatives based on Hamming distance.
     - Update: Recompute the mode of each cluster (i.e. each feature should have most common response).
- A mixture of categorical and numerical features uses K-prototype algorithm (i.e. combination of K-means and K-modes algorithm).

Kernel K-Means

- K-Means algorithm cannot be used to detect non-convex clusters.
  - It can only detect clusters that are linearly separable (convex shapes).
  - Convex shaped clusters are those in which the line segment joining any two points lie completely inside the cluster.



Non-Convex shaped clusetrs

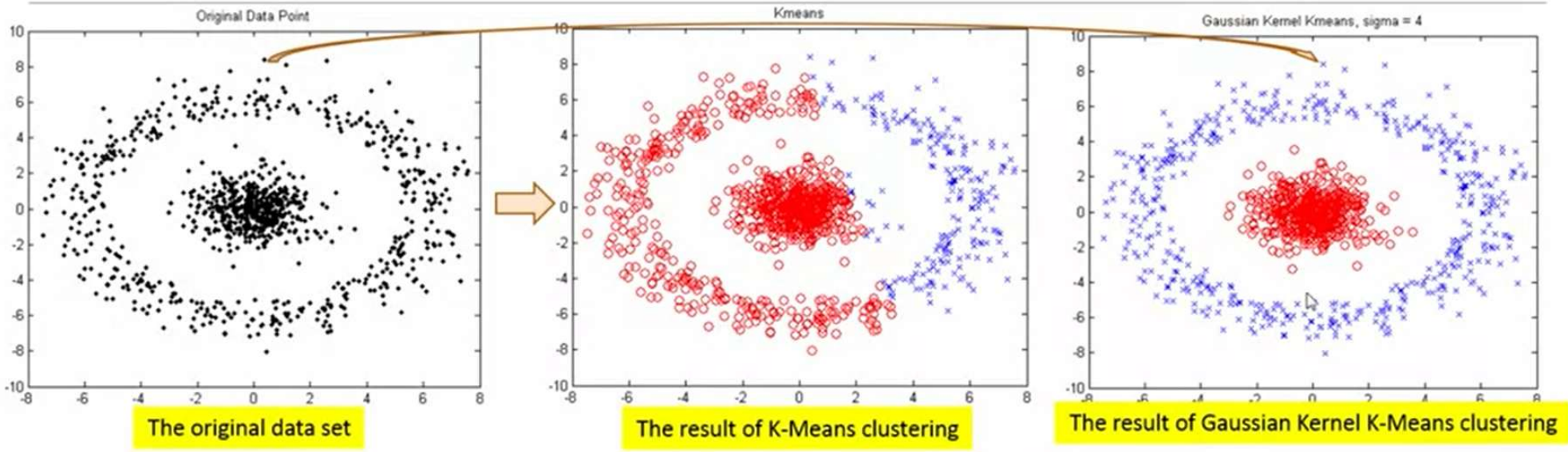


Convex-shaped clusters

- Idea: Project data onto high-dimensional kernel space, and then perform K-means clustering.
  - Map data points in the feature space to a high-dimensional feature space using Kernel Functions.
  - Apply K-means clustering on the mapped feature space.

Kernel K-Means (Contd….)

- Computational complexity is higher than K-Means.
  - Need to compute and store the nxn kernel matrix generated from the kernel function on original data (of order kxn).



Some Kernel Functions

$$Gaussian\ Radial\ Bais\ Function = K(x_i, x_j) = e^{-\left(\frac{|x_i - x_j|^2}{2\sigma^2}\right)}$$

$$Linear\ Kernel = K_{ij} = x_i . x_j$$

$$Polynomial\ Kernel = K_{ij} = (x_i . x_j + constant)^{degree}$$

$$Sigmoid\ Kernel = K_{ij} = \tanh(a(x_i . x_j) + \mathrm{b}) \text{ for constants a, b}$$

$$Log\ Kernel = K_{ij} = -\log(|x_i - x_j|^{degree}) + 1$$

Kernel K-Means Example

- Gaussian radial basis function (RBF) kernel: $K(\boldsymbol{X}_i, \boldsymbol{X}_j) = e^{-\|X_i - X_j\|^2 / 2\sigma^2}$
- Suppose there are 5 original 2-dimensional points:
  - $x_1(0, 0)$, $x_2(4, 4)$, $x_3(-4, 4)$, $x_4(-4, -4)$, $x_5(4, -4)$
- If we set σ to 4, we will have the following points in the kernel space
  - E.g., $\left\|x_1 - x_2\right\|^2 = (0-4)^2 + (0-4)^2 = 32$, therefore, $K(x_1, x_2) = e^{-\frac{32}{2 \cdot 4^2}} = e^{-1}$

Original Space

	x	y
x_1	0	0
x_2	4	4
x_3	-4	4
x_4	-4	-4
x_5	4	-4

RBF Kernel Space ($\sigma = 4$)

$K(x_l, x_1)$	$K(x_l, x_2)$	$K(x_l, x_3)$	$K(x_l, x_4)$	$K(x_l, x_5)$
0	$e^{-\frac{4^2+4^2}{2 \cdot 4^2}} = e^{-1}$	e^{-1}	e^{-1}	e^{-1}
e^{-1}	0	e^{-2}	e^{-4}	e^{-2}
e^{-1}	e^{-2}	0	e^{-2}	e^{-4}
e^{-1}	e^{-4}	e^{-2}	0	e^{-2}
e^{-1}	e^{-2}	e^{-4}	e^{-2}	0