2.2 Clustering Algorithm

Cluster analysis is the assignment of a set of observations into subsets (called *clusters*) so that observations within the same cluster are similar according to some predesignated criterion or criteria, while observations drawn from different clusters are dissimilar. Different clustering techniques make different assumptions on the structure of the data, often defined by some *similarity metric* and evaluated for example by *internal compactness* (similarity between members of the same cluster) and *separation* between different clusters. Other methods are based on *estimated density* and *graph connectivity*. Clustering is a method of unsupervised learning, and a common technique for statistical data analysis. The most common clustering algorithm is k-means algorithm and k-nearest neighborhood.

2.2.1 K-means Clustering Algorithm

k-means is one of the simplest unsupervised learning algorithms that solve the well-known clustering problem. The procedure follows a simple and easy way to classify a given data set through a certain number of clusters (assume k clusters) fixed a priori. The main idea is to define k centers, one for each cluster. These centers should be placed in a cunning way because of different location causes different result. So, the better choice is to place them as much as possible far away from each other. The next step is to take each point belonging to a given data set and associate it to the nearest center. When no point is pending, the first step is completed and an early group age is done. At this point we need to re-calculate k new centroids as barycenter of the clusters resulting from the previous step. After we have these k new centroids, a new binding has to be done between the same data set points and the nearest new center. A loop has been generated. As a result of this loop we may notice

that the ***k*** centers change their location step by step until no more changes are done or in other words centers do not move any more.

Finally, this algorithm aims at minimizing an objective function know as squared error function given by:

$$J(V) = \sum_{i=1}^{c} \sum_{j=1}^{c_i} (\|x_i - v_i\|)^2$$

where,

$||x_i - v_j||$: is the Euclidean distance between x_i and v_j.

c_i: is the number of data points in i^{th} cluster.

c: is the number of cluster centers.

Algorithmic steps for k-means clustering

Let $X = \{x_1, x_2, \ldots, x_n\}$ be the set of data points and $V = \{v_1, v_2, \ldots, v_c\}$ be the set of centers.

1) Randomly select c cluster centers.
2) Calculate the distance between each data point and cluster centers.
3) Assign the data point to the cluster center whose distance from the cluster center is minimum of all the cluster centers.
4) Recalculate the new cluster center using:

$$v_i = (1/c_i) \sum_{j=1}^{c_i} x_i$$

where, c_i represents the number of data points in i^{th} cluster.

5) Recalculate the distance between each data point and new obtained cluster centers.
6) If no data point was reassigned then stop, otherwise repeat from step 3.

Advantages

1. Fast, robust and easier to understand.
2. Relatively efficient: O(***t k n d***), where ***n*** is # objects, ***k*** is # clusters, ***d*** is # dimension of each object, and ***t*** is # iterations. Normally, ***k, t, d*** << ***n***.
3. Gives best result when data set are distinct or well separated from each other.

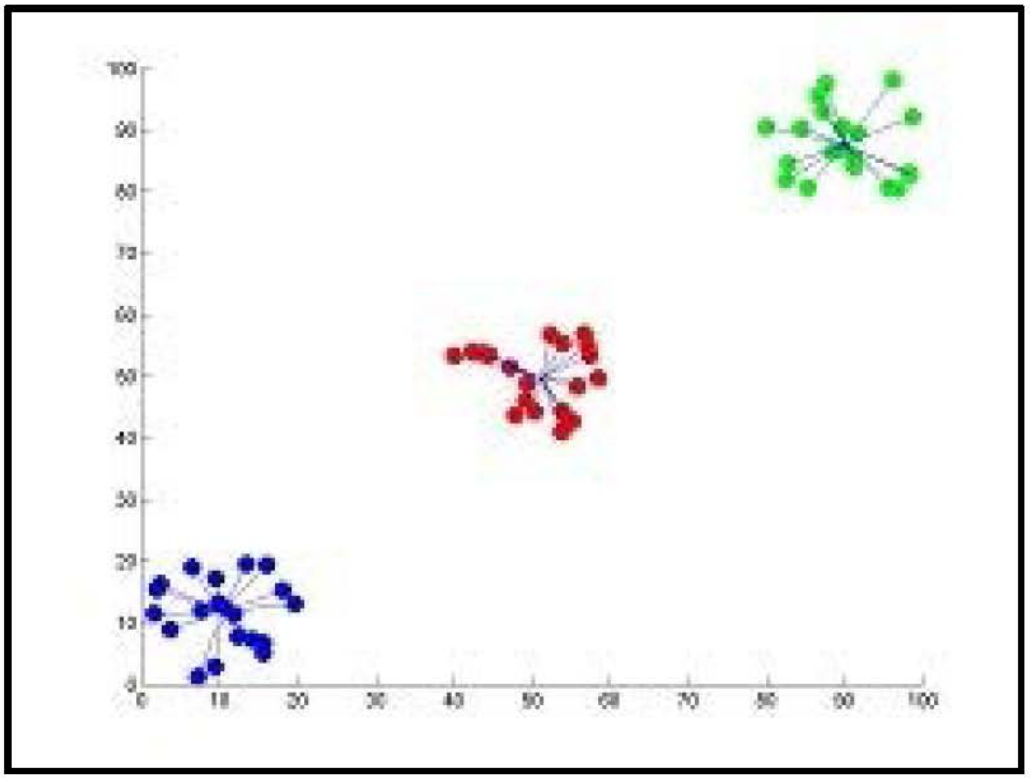


Figure (1) Showing the result of k-means for 'N' = 60 and 'c' = 3

Disadvantages

1. The learning algorithm requires a priori specification of the number of cluster centers.
2. The use of Exclusive Assignment - If there are two highly overlapping data then k-means will not be able to resolve that there are two clusters.
3. The learning algorithm is not invariant to non-linear transformations i.e. with different representation of data we get different results (data represented in form of Cartesian coordinates and polar coordinates will give different results).
4. Euclidean distance measures can unequally weight underlying factors.
5. The learning algorithm provides the local optima of the squared error function.
6. Randomly choosing of the cluster center cannot lead us to the fruitful result.
7. Applicable only when mean is defined i.e. fails for categorical data.
8. Unable to handle noisy data and outliers.
9. Algorithm fails for non-linear data set

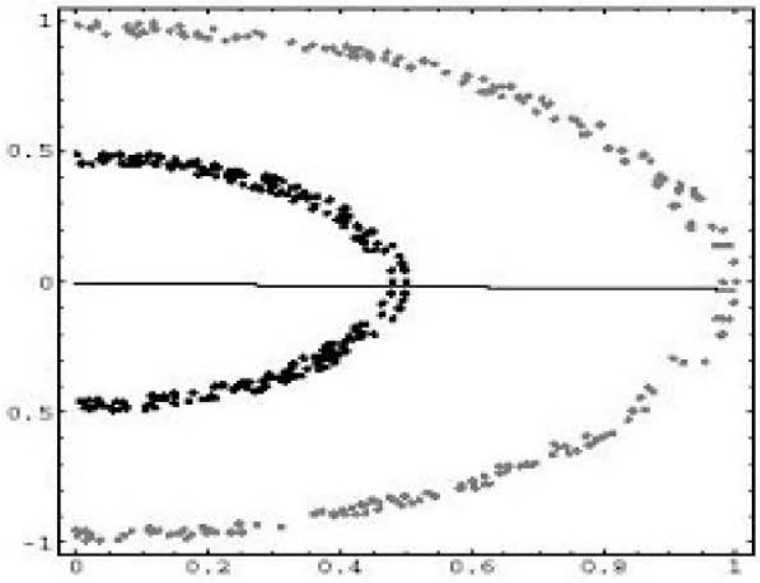


Figure (2) Showing the non-linear data set where *k*-means algorithm fails.

Example: K-Means Clustering Algorithm:

We have height and weight information. Using these two variables, we need to group the objects based on height and weight information.

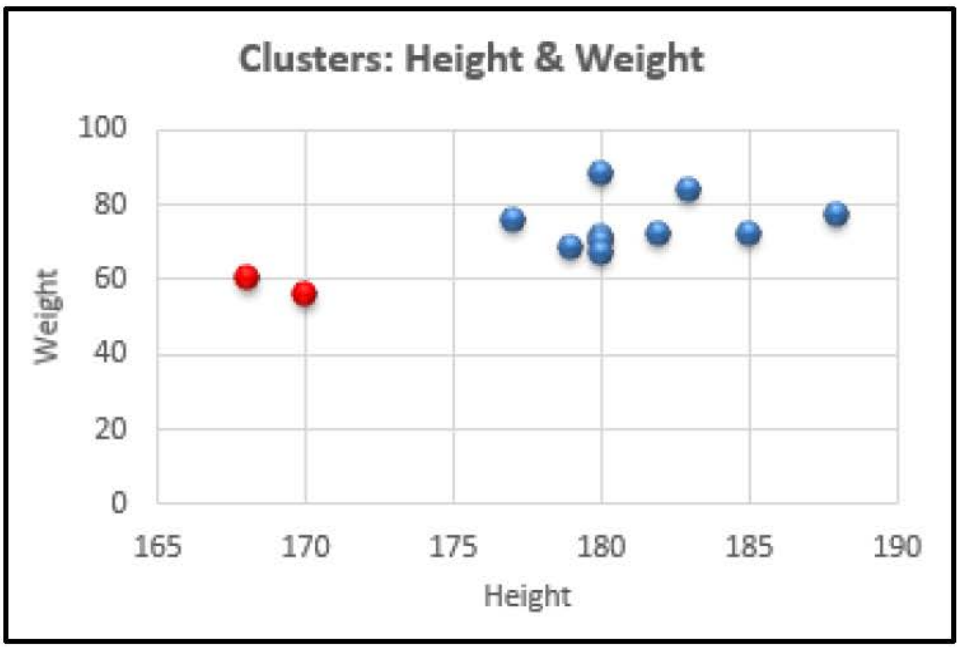


Figure (3) Data Scattering

From the above figure, there are two visible clusters/segments and we want these to be identified using K Means algorithm.

Data Sample

Height	Weight
185	72
170	56
168	60
179	68
182	72
188	77
180	71
180	70
183	84
180	88
180	67
177	76

Step 1: Input

Dataset, Clustering Variables and Maximum Number of Clusters (K in Means Clustering) In this dataset, only two variables –height and weight – are considered for clustering

Height	Weight
185	72
170	56
168	60
179	68
182	72
188	77
180	71
180	70
183	84
180	88
180	67
177	76

Step 2: Initialize cluster centroid

In this example, the value of K is considered as 2. Cluster centroids are initialized with first 2 observations.

Cluster	Initial Centroid	
	Height	Weight
K=1	185	72
K=2	170	56

Step 3: Calculate Euclidean Distance

Euclidean is one of the distance measures used on K Means algorithm.

Euclidean Distance = $\sqrt{(X_H - H_1)^2 + (X_W - W_1)^2}$

Where

X_H: Observation value of variable Height

H_1: Centroid value of Cluster 1 for variable Height

X_W: Observation Value of variable Weight

W_1: Centroid value of cluster 1 for variable Weight

The Euclidean distance between the observation and initial cluster centroids 1 and 2 is calculated. Based on Euclidean distance each observation is assigned to one of the clusters - based on minimum distance.

First two observations:

Height	Weight
185	72
170	56

Now initial cluster centroids are:

Cluster	Update Centroid	
	Height	Weight
K=1	185	72
K=2	170	56

Euclidean Distance Calculation from each of the clusters is calculated.

Euclidian Distance from Cluster 1	Euclidian Distance from Cluster 2	Assignments
$(185-185)^2+(72-72)^2=0$	$(185-170)^2+(72-56)^2= 21.93$	1
$(170-185)^2+(56-72)^2= 21.93$	$(170-170)^2+(56-56)^2= 0$	2

We have considered two observations for assignment only because we knew the assignment. And there is no change in Centroids as these two observations were only considered as initial centroids

Step 4: Move on to next observation and calculate Euclidean Distance:

Height	Weight
168	60

Euclidian Distance from Cluster 1	Euclidian Distance from Cluster 2	Assignments
$(168-185)^2+(60-72)^2=20.808$	$(168-170)^2+(60-72)^2=20.808$	2

Since distance is minimum from cluster 2, so the observation is assigned to cluster 2. Now revise Cluster Centroid – mean value Height and Weight as Custer Centroids. Addition is only to cluster 2, so centroid of cluster 2 will be updated

Updated cluster centroids:

Cluster	Update Centroid	
	Height	Weight
K=1	185	72
K=2	(170+168)/2=169	(56+60)/2=58

Step 5: Calculate Euclidean Distance for the next observation, assign next observation based on minimum Euclidean distance and update the cluster centroids.

Next Observation:

Height	Weight
179	68

Euclidean Distance Calculation and Assignment:

Euclidian Distance from Cluster 1	Euclidian Distance from Cluster 2	Assignments
7.211103	14.14214	2

Update Cluster Centroid:

Cluster	Update Centroid	
	Height	Weight
K=1	182	70.6667
K=2	169	58

Continue the steps until all observations are assigned

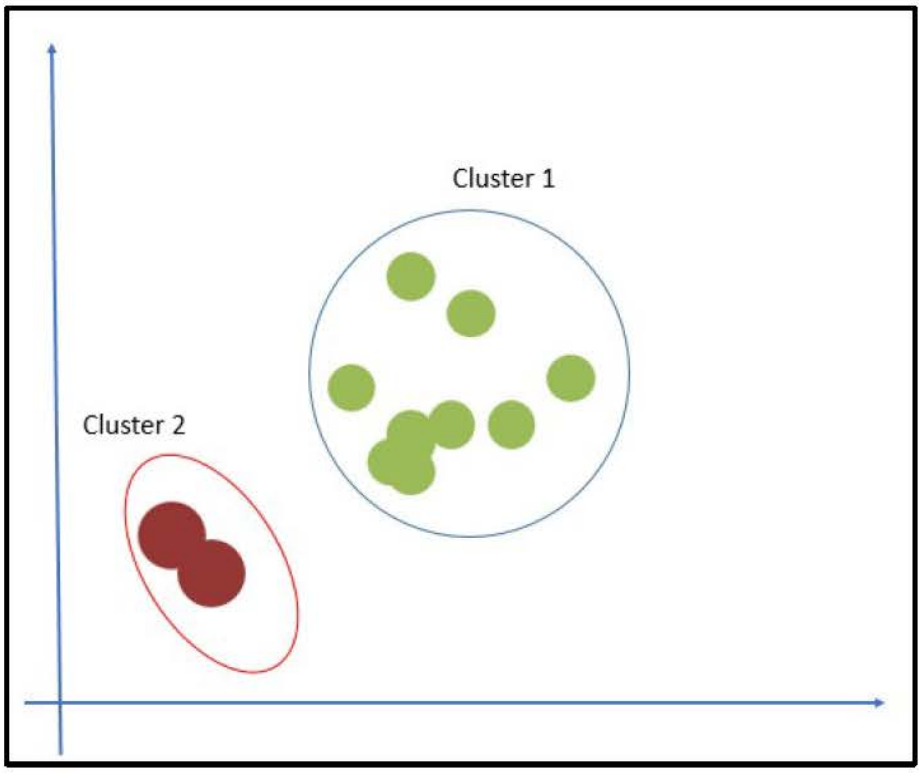
Cluster Centroids

Cluster	Centroid	
	Height	Weight
K=1	182.8	72
K=2	169	58

Final assignments:

Height	Weight	Assignment
185	72	1
170	56	2
168	60	2
179	68	1
182	72	1
188	77	1
180	71	1
180	70	1
183	84	1
180	88	1
180	67	1
177	76	1

This is what was expected initially based on two-dimensional plot



A few important considerations in K Means

- Scale of measurements influences Euclidean Distance, so variable standardization becomes necessary.
- Depending on expectations - you may require outlier treatment
- K Means clustering may be biased on initial centroids - called cluster seeds
- Maximum clusters is typically inputs and may also impacts the clusters getting created In the next blog, we focus on creating clusters using R. K Means Clustering using R