Clustering

Earning is in Learning
- Rajesh Jakhotia

Content

- Clustering Definition
- Distance Measure
- Hierarchical Clustering
- K Mean Clustering

Learning Objectives

greatlearning

- Why Clustering?
- What is Clustering?
- Various Distance Measures
- Hierarchical Clustering
- K Means Clustering

Clustering Definitions
Distance Measures

Why Clustering? Applications of Clustering

- Why Clustering?
  - To group similar objects / data points
  - To find homogenous sets of customers
  - To segment the data in similar groups

- Applications:
  - Marketing : Customer Segmentation & Profiling
  - Libraries : Book classification
  - Retail : Store Categorization

What is Clustering?

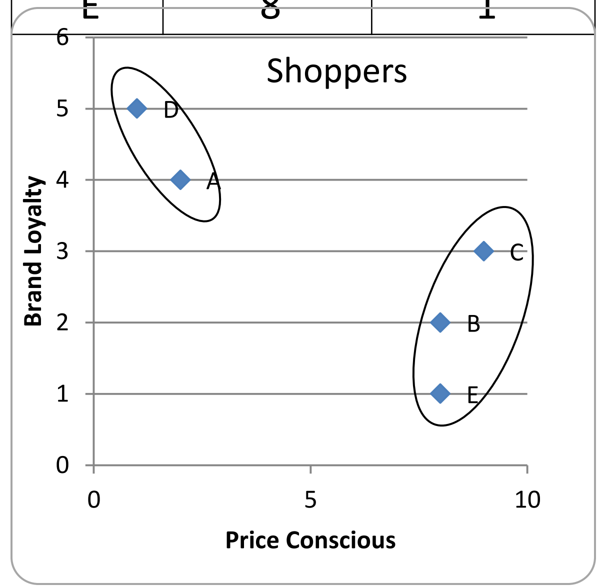
- Clustering is a technique for finding similar groups in data, called clusters.
- Clustering is an Unsupervised Learning Technique
- Clustering can also be thought of as a case reduction technique wherein it groups together similar records in cluster
- Clustering helps simplify data by reducing many data points into a few clusters (segments)

What is a Cluster?

- A *cluster* can be defined as a collection of objects which are "similar" between them and are "dissimilar" to the objects belonging to other clusters

- How do we define "Similar" in clustering?
  - Based on Distance

Shoppers	Price Conscious	Brand Loyalty
A	2	4
B	8	2
C	9	3
D	1	5
E	8	1



greatlearning

How do we define "(dis) Similar" ?

- Similar in clustering is based on Distance
- Various distance measures
  - Eucledian Distance
  - Chebyshev Distance
  - Manhattan Distance ... and more

A

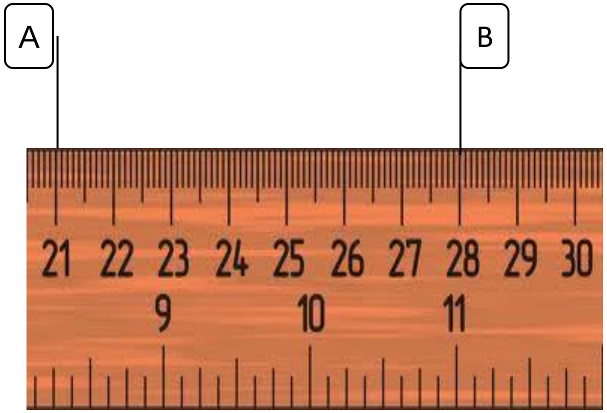
Block — Manhattan Distance = 8 + 4 = 12

Block — Chebyshev Distance = Max (8, 4) = 8

Block — Eucledian Distance = sqrt (8^2 + 4^2) = 8.94

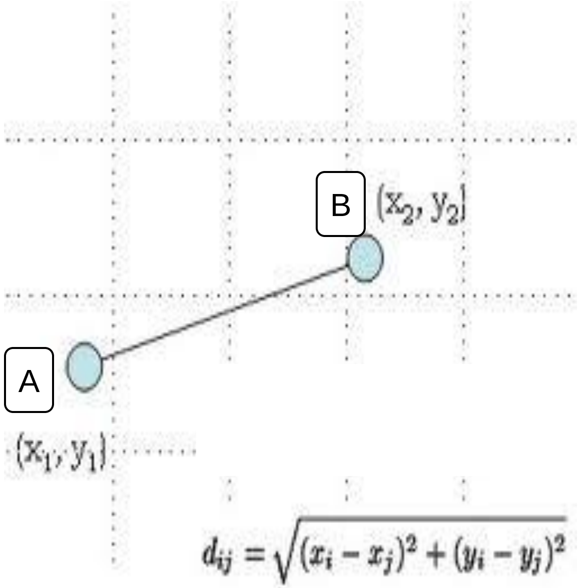
Block Block Block Block Block Block Block Block B

Distance Computation



What is the distance between Point A and B?

Ans: 7



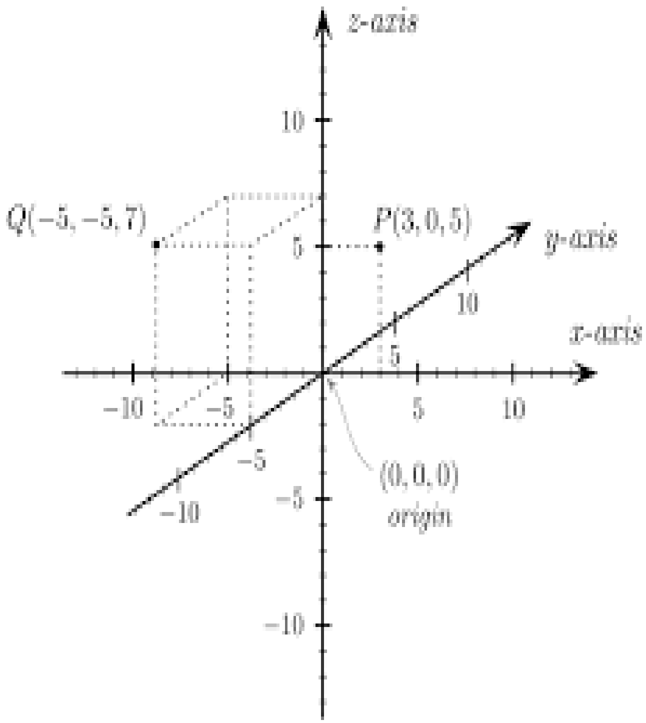
What is the distance between Point A and B?

Ans:

$$\sqrt{[\,(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

(Remember the Pythagoras Theorem)

Eucledian Distance



- What is the distance between Point A and B in n-Dimension Space?
- If A $(x_1, y_1, \ldots Z_1)$ and B $(x_2, y_2, \ldots z_2)$ are cartesian coordinates
- By using **Euclidean Distance** we get Distance AB as
- $D_{AB} = \sqrt{[(x_2-x_1)^2 + (y_2-y_1)^2 + \ldots + (z_2-z_1)^2]}$

Chebyshev Distance

greatlearning

- In mathematics, **Chebyshev distance** is a metric defined on a vector space where the distance between two vectors is the greatest of their differences along any coordinate dimension

- Assume two vectors: A $(x_1, y_1, \ldots\ z_1)$ & B $(x_2, y_2, \ldots\ z_2)$

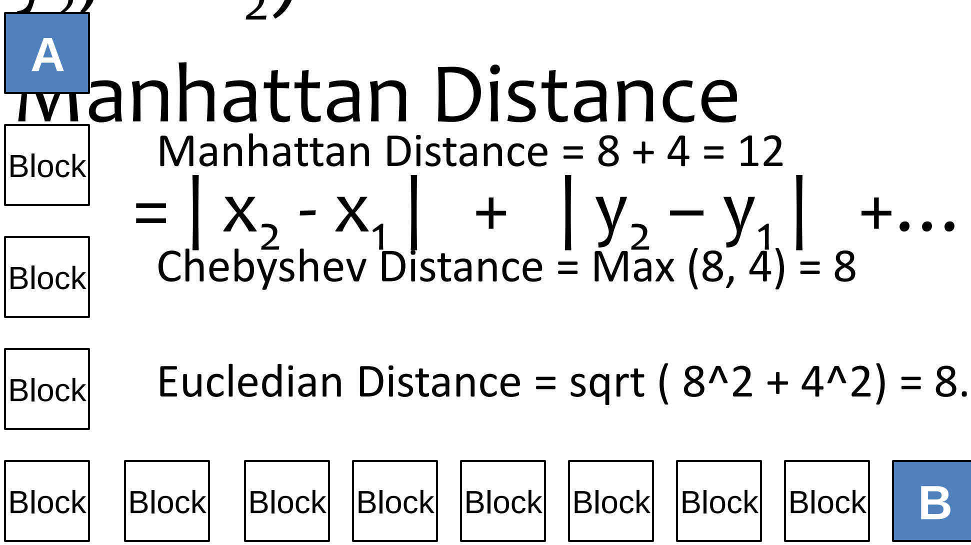
Reference Link : https://en.wikipedia.org/wiki/Chebyshev_distance

- Chebyshev Distance

Manhattan Distance

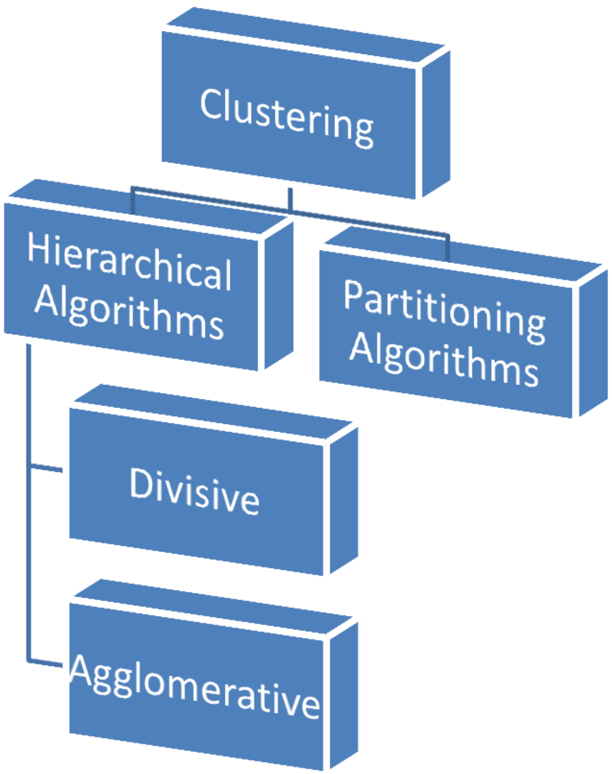
- Manhattan Distance also called City Block Distance
- Assume two vectors: A $(x_1, y_1, \ldots z_1)$ & B $(x_2, y_2, \ldots z_2)$
- Manhattan Distance

$$= |x_2 - x_1| + |y_2 - y_1| + \ldots\ldots |z_2 - z_1|$$



Types of Clustering

Types of Clustering Procedures



- **Hierarchical clustering** is characterized by a tree like structure and uses distance as a measure of (dis)similarity

- **Partitioning Algorithms** starts with a set of partitions as clusters and iteratively refines the partitions to form stable clusters

Steps involved in Clustering

Formulate the problem – Select variables to be used for clustering

↓

Decide the Clustering Procedure (Hierarchical / Partitioning)

↓

Select the measure of similarity (dis-similarity)

↓

Choose cluster linkage algorithm (applicable in hierarchical clustering)

↓

Decide on the number of clusters

↓

Interpret the cluster output (Profile the clusters)

↓

Validate the clusters

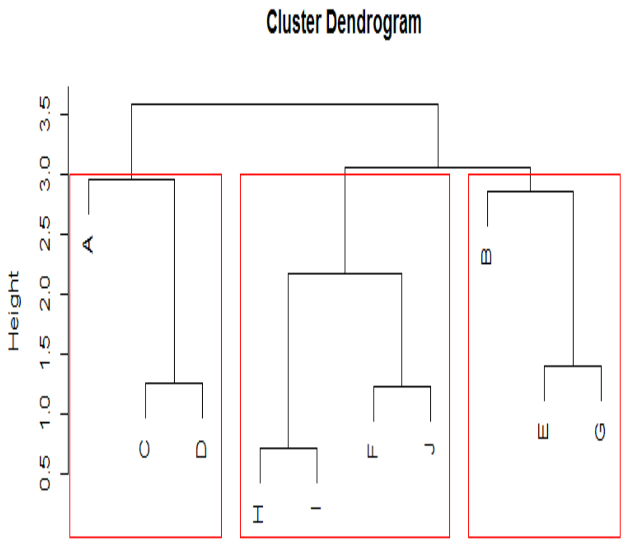
Hierarchical Clustering

Hierarchical Clustering

- Hierarchical Clustering is a clustering techniques which tends to create clusters in a hierarchical tree like structure

- Hierarchical clustering makes use of Distance as a measure of similarity

- Cluster tree like output is called Dendogram

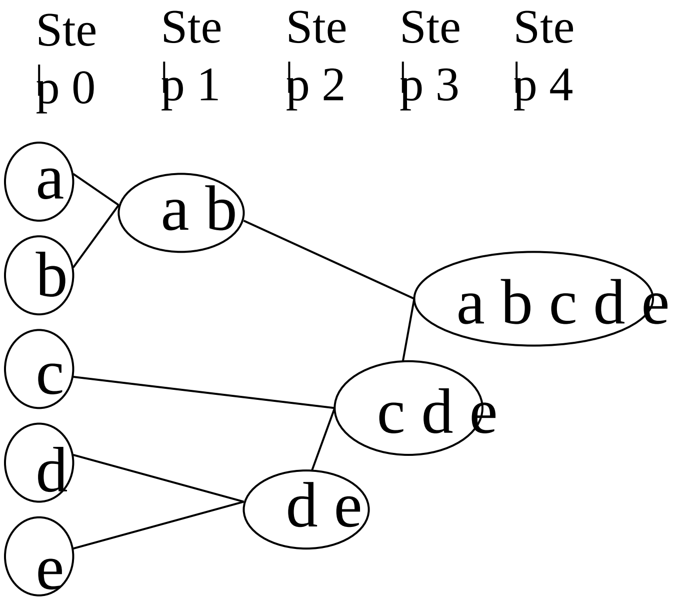


Hierarchical Clustering | Agglomerative Clustering Steps

- Starts with each record as a cluster of one record each

- Sequentially merges 2 closest records by distance as a measure of (dis)similarity to form a cluster. This reduces the number of records by 1

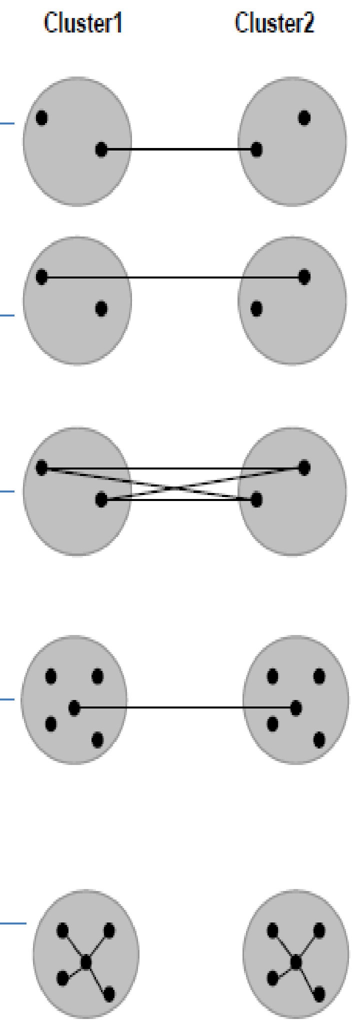
- Repeat the above step with new cluster and all remaining clusters till we have one big cluster



How do you measure the distance between cluster (a,b) and (c) or the cluster (a,b) and (d,e)
????

Agglomerative Clustering Linkage Algorithms

- Single linkage – Minimum distance or Nearest neighbour rule
- Complete linkage – Maximum distance or Farthest distance
- Average linkage – Average of the distances between all pairs
- Centroid method – combine cluster with minimum distance between the centroids of the two clusters
- Ward’s method – Combine clusters with which the increase in within cluster variance is to the smallest degree



Hierarchical Clustering for Retail Customers

Let us find the clusters in given Retail Customer Spends data
We will use Hierarchical Clustering technique
Let us first set the working directory path and import the data

```
setwd ("D:/K2Analytics/Clustering/")
RCDF <- read.csv("datafiles/Cust_Spend_Data.csv", header=TRUE)
View(RCDF)
```

Cust_ID	Name	Avg_Mthly_Spend	No_Of_Visits	Apparel_Items	FnV_Items	Staples_Items
1	A	10000	2	1	1	0
2	B	7000	3	0	10	9
3	C	7000	7	1	3	4
4	D	6500	5	1	1	4
5	E	6000	6	0	12	3
6	F	4000	3	0	1	8
7	G	2500	5	0	11	2
8	H	2500	3	0	1	1
9	I	2000	2	0	2	2
10	J	1000	4	0	1	7

HyperMarket Customer Spend MetaData

AVG_Mthly_Spend: The average monthly amount spent by customer

No_of_Visits: The number of times a customer visited the HyperMarket in a month

Item Counts: Count of **Apparel, Fruits and Vegetable, Staple Items** purchased in a month

Building the hierarchical clusters (without variable scaling)

?dist ## to get help on distance function

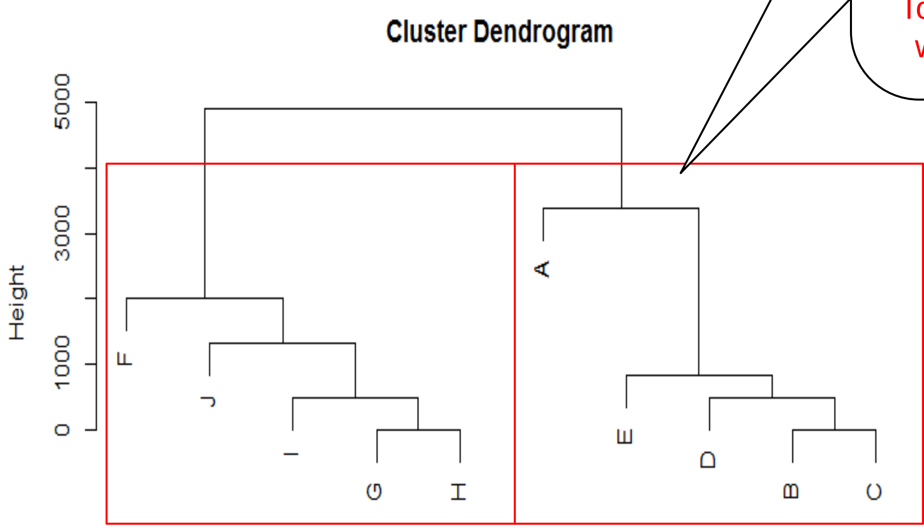
d.euc <- dist(x=RCDF[,3:7], method = "euclidean")

we will use the hclust function to build the cluster

?hclust ## to get help on hclust function

clus1 <- hclust(d.euc, method = "average")

plot(clus1, labels = as.character(RCDF[,2]))



Note: The two clusters formed are primarily on the basis of AVG_MTHLY_SPEND

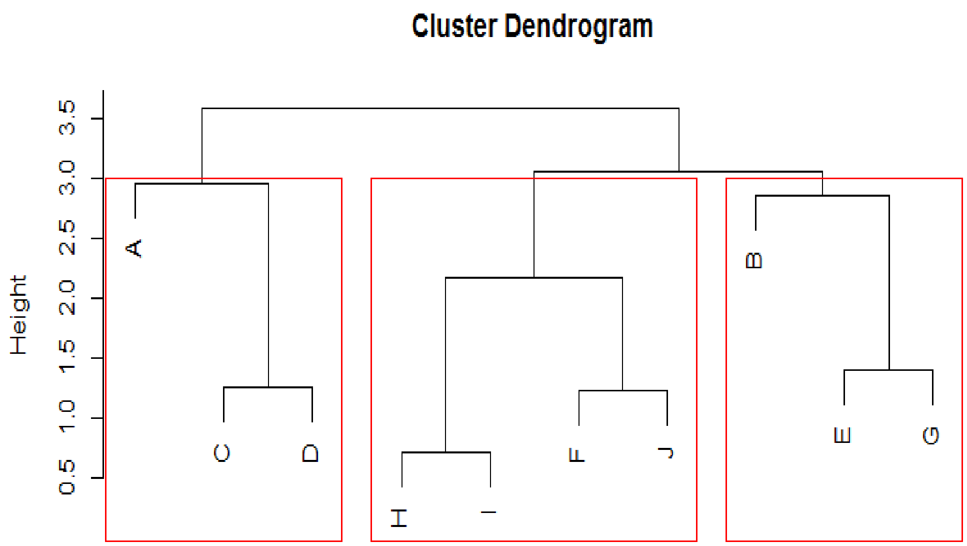
Eucledian Distance computation in this case is influenced by AVG_MTHLY_SPEND variable as the range of this variable is too large compared to the other variables

To avoid this problem, we should scale the variables used for clustering

Building the hierarchical clusters (with variable scaling)

scale function standardizes the values

```
scaled.RCDF <- scale(RCDF[,3:7])
head(scaled.RCDF, 10)
d.euc <- dist(x=scaled.RCDF, method = "euclidean")
clus2 <- hclust(d.euc, method = "average")
plot(clus2, labels = as.character(RCDF[,2]))
rect.hclust(clus2, k=3, border="red")
```



Understanding the Height Calculation in Clustering

Let us see the distance matrix

d

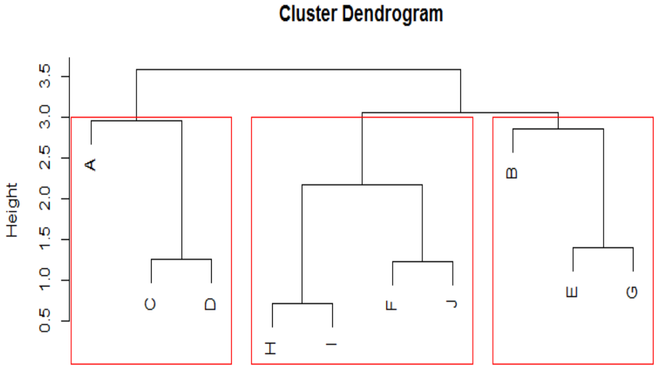
Dist.	A	B	C	D	E	F	G	H	I
B	4.25								
C	3.41	3.84							
D	2.51	3.47	1.26						
E	4.27	2.70	2.92	3.20					
F	3.98	2.21	3.58	2.85	3.43				
G	4.38	3.02	3.38	3.35	1.41	3.17			
H	3.40	3.60	3.66	2.93	3.24	2.35	2.46		
I	3.53	3.39	4.05	3.21	3.48	2.18	2.61	0.73	
J	4.55	2.97	3.59	3.04	3.41	1.24	2.80	2.12	2.06

Let us see the height for clusters

clus2$height

```
[1] 0.7272685 1.2410400 1.2635399 1.4064486 2.1742679 2.8590372 2.9615235 3.0647590 3.5925234
```

ning



Dist.	A	B	C	D	E	F	G	H	I
B	4.25								
C	3.41	3.84							
D	2.51	3.47	1.26						
E	4.27	2.70	2.92	3.20					
F	3.98	2.21	3.58	2.85	3.43				
G	4.38	3.02	3.38	3.35	1.41	3.17			
H	3.40	3.60	3.66	2.93	3.24	2.35	2.46		
I	3.53	3.39	4.05	3.21	3.48	2.18	2.61	0.73	
J	4.55	2.97	3.59	3.04	3.41	1.24	2.80	2.12	2.06

C, D	1.26

H, I	0.73	F, J	1.24

E, G	1.41

A, (C,D)	
A, C	3.41
A, D	2.51
A, (C,D)	2.96

(H,I), (F,J)		
	F	J
H	2.35	2.12
I	2.18	2.06
((H,I), (F,J))		2.17

B, (E,G)	
B, E	2.70
B, G	3.02
B, (E,G)	2.86

(H,I, F,J) , (B, E,G)			
	B	E	G
H	3.60	3.24	2.46
I	3.39	3.48	2.61
F	2.21	3.43	3.17
J	2.97	3.41	2.80
(H,I, F,J) , (B, E,G)		3.06	

(A, C, D) , (H, I, F, J, B, E, G)							
	H	I	F	J	B	E	G
A	3.40	3.53	3.98	4.55	4.25	4.27	4.38
C	3.66	4.05	3.58	3.59	3.84	2.92	3.38
D	2.93	3.21	2.85	3.04	3.47	3.20	3.35
(A, C, D) , (H, I, F, J, B, E, G)					3.59		

Profiling the clusters

```
## profiling the clusters
RCDF$Clusters <- cutree(clus2, k=3)
aggr = aggregate(RCDF[, -c(1,2, 8)], list(RCDF$Clusters), mean )
clus.profile <- data.frame( Cluster = aggr[,1]  ,
                    Freq = as.vector(table(RCDF$Clusters)) ,
                    aggr[,-1]
                 )
View(clus.profile)
```

Cluster	Freq	Avg_Mthly_Spend	No_Of_Visits	Apparel_Items	FnV_Items	Staples_Items
1	3	7833.333	4.666667	1	1.666667	2.666667
2	3	5166.667	4.666667	0	11.000000	4.666667
3	4	2375.000	3.000000	0	1.250000	4.500000

Partitioning Clustering

K Means Clustering

K Means Clustering

- K-Means is the most used, non-hierarchical clustering technique
- It is not based on Distance…
- It is based on within cluster Variation, in other words Squared Distance from the Centre of the Cluster
- The algorithm aims at segmenting data such that within cluster variation is reduced

K Means Algorithm

- Input Required : No of Clusters to be formed. (Say K)

- Steps
  1. Assume K Centroids (for K Clusters)
  2. Compute Eucledian distance of each objects with these Centroids.
  3. Assign the objects to clusters with shortest distance
  4. Compute the new centroid (mean) of each cluster based on the objects assigned to each clusters. The K number of means obtained will become the new centroids for each cluster
  5. Repeat step 2 to 4 till there is convergence
     - i.e. there is no movement of objects from one cluster to another
     - Or threshold number of iterations have occurred

K-means advantages

- K-means is superior technique compared to Hierarchical technique as it is less impacted by outliers

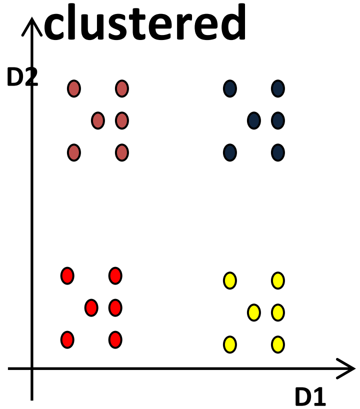
- Computationally it is more faster compared to Hierarchical

- Preferable to use on interval or ratio-scaled data as it uses Eucledian distance… desirable to avoid using on ordinal data

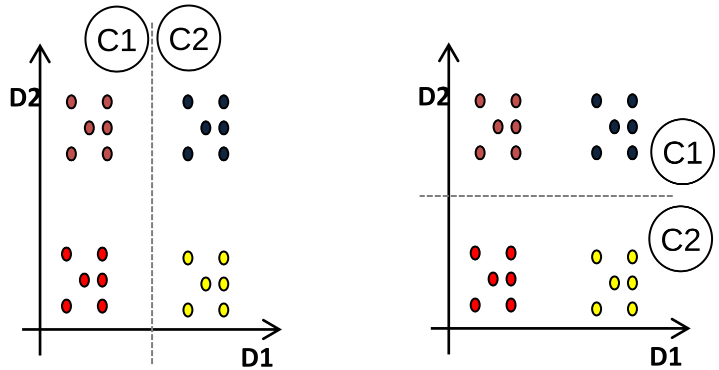
- **Challenge – Number of clusters are to be pre-defined and to be provided as input to the process**

Why find optimal No. of Clusters?

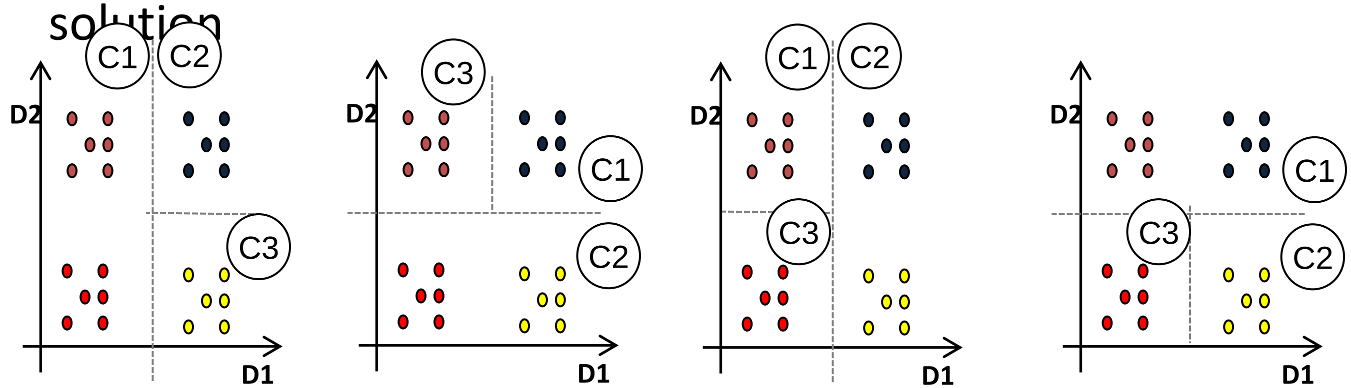
Data to be clustered



- Two Clusters – 2 possible solution

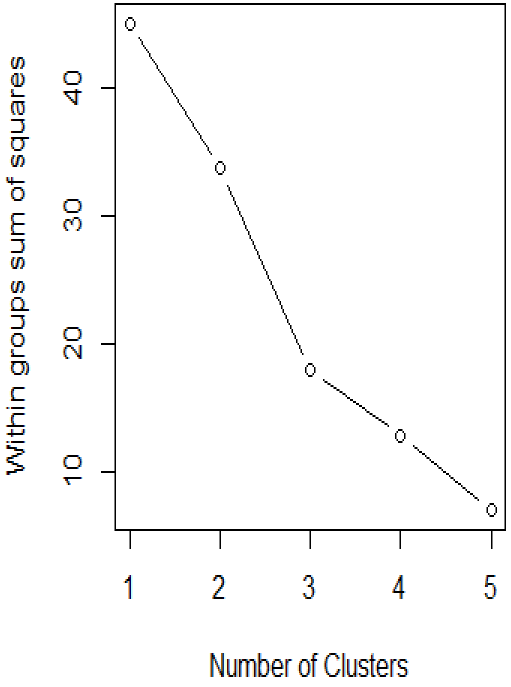


- Three Clusters – Multiple possible solution



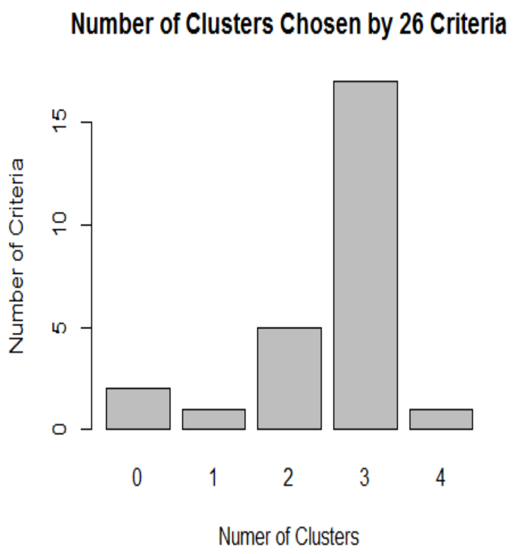
R code to get Optimal No. of Clusters

```
## code taken from the R-statistics blog  http://www.r-statistics.com/2013/08/k-means-clustering-from-r-in-action/
## Identifying the optimal number of clusters form WSS
wssplot <- function(data, nc=15, seed=1234) {
  wss <- (nrow(data)-1)*sum(apply(data,2,var))
  for (i in 2:nc) {
    set.seed(seed)
    wss[i] <- sum(kmeans(data, centers=i)$withinss)}
  plot(1:nc, wss, type="b", xlab="Number of Clusters",
       ylab="Within groups sum of squares")}

wssplot(scaled.RCDF, nc=5)
```



Using NbClust to get optimal No. of Clusters

```
## Identifying the optimal number of clusters
## install.packages("NbClust")
library(NbClust)
set.seed(1234)
nc <- NbClust(KRCDF[,c(-1,-2)], min.nc=2, max.nc=4, method="kmeans")
table(nc$Best.n[1,])

barplot(table(nc$Best.n[1,]),
        xlab="Numer of Clusters", ylab="Number of Criteria",
        main="Number of Clusters Chosen by 26 Criteria")
```



K Means Clustering R Code

```
?kmeans
kmeans.clus = kmeans(x=scaled.RCDF, centers = 3, nstart = 25)
## x = data frame to be clustered
## centers = No. of clusters to be created
## nstart = No. of random sets to be used for clustering
kmeans.clus
```

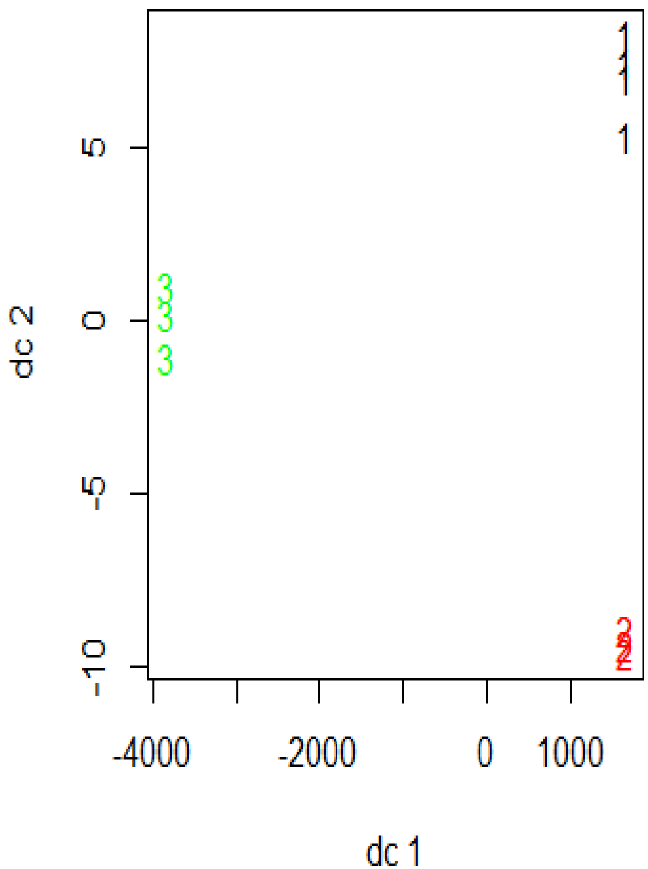
```
K-means clustering with 3 clusters of sizes 4, 3, 3

Cluster means:
  Avg_Mthly_Spend No_Of_Visits Apparel_Items  FnV_Items Staples_Items
1      -0.8600931   -0.5883484     -0.621059 -0.6500980     0.1636634
2       1.0367452    0.3922323      1.449138 -0.5612868    -0.4364358
3       0.1100456    0.3922323     -0.621059  1.4280842     0.2182179

Clustering vector:
 [1] 2 3 2 2 3 1 3 1 1 1

Within cluster sum of squares by cluster:
[1] 5.256752 6.514105 6.126217
 (between_SS / total_SS =  60.2 %)

Available components:

[1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss" "betweenss"
[7] "size"         "iter"         "ifault"
```

Plotting the clusters

plotting the clusters
install.packages("fpc")

library(fpc)

plotcluster(scaled.RCDF, kmeans.clus$cluster)

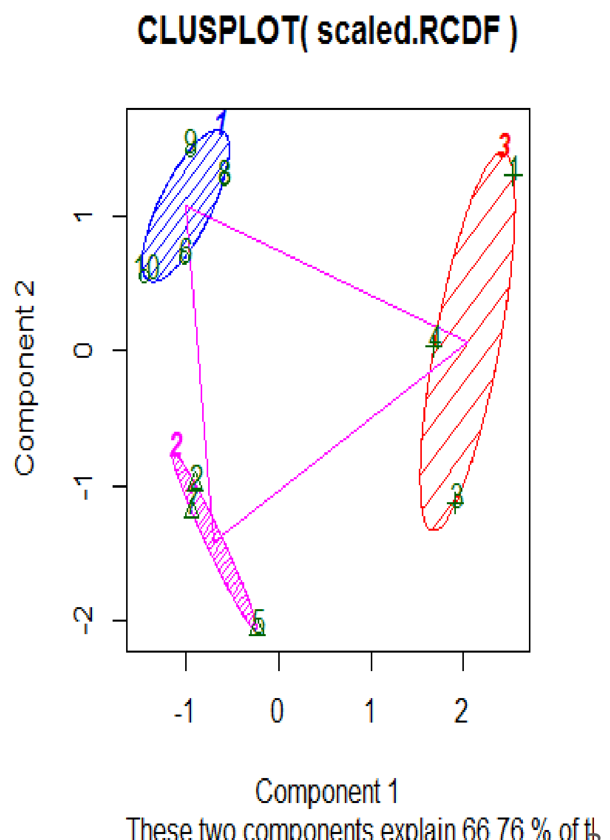


plotting the clusters

install.packages("fpc")

library(fpc)

plotcluster(scaled.RCDF, kmeans.clus$cluster)



Profiling the clusters

```
## profiling the clusters
KRCDF$Clusters <- kmeans.clus$cluster
aggr = aggregate(KRCDF[,-c(1,2, 8)],list(KRCDF$Clusters),mean)
clus.profile <- data.frame( Cluster=aggr[,1],
                 Freq=as.vector(table(KRCDF$Clusters)),
                 aggr[,-1])

View(clus.profile)
```

Cluster	Freq	Avg_Mthly_Spend	No_Of_Visits	Apparel_Items	FnV_Items	Staples_Items
1	4	2375.000	3.000000	0	1.250000	4.500000
2	3	7833.333	4.666667	1	1.666667	2.666667
3	3	5166.667	4.666667	0	11.000000	4.666667

Next steps after clustering

- Clustering provides you with clusters in the given dataset
- Clustering does not provide you rules to classify future records
- To be able to classify future records you may do the following
  - Build Discriminant Model on Clustered Data
  - Build Classification Tree Model on Clustered Data

References

- Chapter 9 : Cluster Analysis (http://www.springer.com)
  - Google search : "www.springer.com cluster analysis chapter 9"

- http://sites.stat.psu.edu/~ajw13/stat505/fa06/19_cluster/09_cluster_wards.html

- https://home.deib.polimi.it/matteucc/Clustering/tutorial_html/

Thank you

Contact us:

ar.jakhotia@k2analytics.co.in