

INDIAN INSTITUTE OF TECHNOLOGY ROORKEE
NPTEL
NPTEL ONLINE CERTIFICATION COURSE
Business Analytics & Data Mining Modeling
Using R – Part II
Lecture-07
ClusterAnalysis– Part III
With
Dr. Gaurav Dixit
Department of Management Studies
Indian Institute of Technology Roorkee

Business Analytics & Data Mining Modeling Using R - Part II

Lecture-07 Cluster Analysis-Part III



With
Dr. Gaurav Dixit
Department of Management Studies
Indian Institute of Technology Floorkee

Welcome to the course Business Analytics and Data Mining Modeling Using R – Part 2, so in previous few lectures we have started our discussion on cluster analysis, and specifically we were discussing the distance metrics for categorical data, so let's continue our discussion, so we talked about two specific you know coefficient, matching coefficient and Jaquard's coefficient and how they can be used to measure categorical variables, so then we also talked about the

Cluster Analysis

- Distance Metrics for categorical data

$$\text{matching coefficient} = \frac{a + d}{p}$$

$$\text{jaquard's coefficient} = \frac{d}{b + c + d}$$

- Jaquard's coefficient ignores absence of variables
 - Since presence of an attribute in both the observations can be taken as evidence of similarity, however same cannot be said about absence of an attribute (it adds uncertainty)



situation, the scenario where both numerical variables and categorical variables are present, so in such a scenario how we can you know, what are the metrics which could be used to measure distance.

Cluster Analysis

- Distance Metrics for both numerical and categorical data

- Gower's similarity metric can be used

- To compute this metric, first all the variables should be scaled to [0,1]
- Then, weighted average of distances is computed using following formula

$$S_{ij} = \frac{\sum_{m=1}^p w_{ijm} s_{ijm}}{\sum_{m=1}^p w_{ijm}}$$

- Where $w_{ijm} = 1$ or specified weight if comparison of measurements is valid, else 0
- For binary categorical variables, $s_{ijm} = 1$ for presence of attribute for both the observations, else 0
 - Equivalent to Jaquard's coefficient



So Gower's similarity metric is one that could be used, so let's understand this particular metric, so to compute this metric first all the variables should be scaled to 0, 1, and reason for this is also quite obvious because we are also having a categorical data in our dataset, so there the categorical data it could be ordinary, it could be nominal, and typically we used dummy variables to you know to quote these categorical variables, so therefore those dummy variables typically they are quoted using zeros and ones, and therefore the range is going to be 0 and 1, so

for that reason we typically scale our numerical variables also to this scale, 0, 1 scale, so that is why the first line is asking that, asking for that explicitly, for example first all the variables should be scaled to 0, 1.

The second step would involve taking weighted average of distances computed using following formula, so this formula actually in a sense accommodates both the numerical and categorical variables.

So if we look at this SIJ, so in here VS/SBB, S indicates or denotes the similarity between two observations I and J, so here you can see in the numerator part we have summation over 1 to P, so that is because we have P number of variables, then we have weight value that is WIJM, so where M starts from 1 to P, so this is for each variables, so for each variable we got to have these weights. And then similarity value for these observation, this is again for each variable IJM, so IJ stands for the observations, I and J which are under consideration, and for each variable which ending from 1 to P, so the product of the weight value and the similarity value that is going to be used in the, in a numerator part, so we will take summation of all such values, weight and product of, weight and similarity value, and in the denominator we are using the summation of you know weight values, right.

So let's understand how these values are going to be computed, so first of all let's understand the WIJM the weight part, so WIJM is going to be 1 or specified weight, so if we are going for you know equal weightage scenario where we are assigning equal weightage to all the variables, then of course the typically value is going to be 1, otherwise if he have different weightage for different variables, then that specified weight value can actually be used, so typically the WIJM value is going to be 1 or a specified weight you know if comparison of measurements is valid, otherwise 0, so if the measurements that is specifically we are talking about the similarity value, if that similarity value can be compare for both the observation of I and J, then we are going to use this one part, otherwise 0.

So let's move to the next category, so if our variable is binary categorical variable, so SIJM this is going to be 1 for indicating presence of attribute for both the observations else 0, so as we know that in categorical variable this particular discussion is quite similar to Jaquard's coefficient, so one indicating the presence of that attribute for both the observation, and 0 indicating the absence of that attribute for both the observation, so if the, that attribute is present for you know both the observation then we'll take SIJM value as 1, if it not present then we'll take it as 0, just like in the Jaquard's coefficient, so there if you know there also in the Jaquard's coefficient if some of the you know variables who are you know absent some of the attributes are absent for both the observations, we did not you know, we did not accounted that as a similarity thing. Similarly here also SIJM is going to be 1, you know that attribute is present for both the observations.

Cluster Analysis

- Distance Metrics for both numerical and categorical data
 - Gower's similarity metric can be used
 - For non-binary categorical variables, $s_{ijm}=1$ for match, else 0
 - For numerical variables,

$$s_{ijm} = 1 - \frac{|x_{im} - x_{jm}|}{\max(x_m) - \min(x_m)}$$

Let's move forward, so if we talk about the non-binary categorical variables, so in that case is so where we have the categorical variable having more than 2 categories, so where we are not using this dummy kind of coding, you know, indicating presence or absence, so those kind of variables can also be covered here, so for non-binary categorical variable SIJM value is going to be 1 for match, so that means two observations, for example here we are talking about observation I and J, and for a particular variable if there are you know three categories and for the you know for different observation if they are, if they are you know, if they have the value, the same attribute for you know same category for that particular variable, for example if we take the you know status of a particular customer whether the customer is employed or student or retired, if two observations are having the same status that is both the observation are having the employed values for this status variable, then in that case match is there, therefore the value SIJM is 1, otherwise if the match is not there then the value is going to be 0.

So the discussion that we had in the previous lecture that if the you know a particular, you know for a particular variable the attributes or the values the categories or matching, then that in a way indicates the similarity, if it is not matching then it indicates the dissimilarity, so because it indicates the similarity, if the value is match, the attribute value match then therefore we are taking this value SIJM value as 1, otherwise 0. So these two point that we just discussed they will cover all the categorical variables, for numerical variables we could use this formula to compute SIJM values, so this formula as you can see 1- then we have this value, in the numerator $|x_{im} - x_{jm}|$ divided by $\max(x_m) - \min(x_m)$, so this particular formula can actually be use, so you can see the covers similarity metric and specifically for numerical variables, the formula that we are using is you know quite similar to, we are using you know absolute differences here, so those absolute differences are being used to compute this value, and then for you know categorical variables and numerical variables, once all these you know similarities value SIJM values have been computed we can plug all these values here, in this formula SIJ formula and we can compute the similarity value for those two observations.

About weight we have already talked about, so in this fashion this particular similarity metric Gower's similarity metric can be used when both kind of variable, continuous variables, and categorical variables are present in the dataset.

So with this till now what we have been able to cover is the distance between two observations, so one important aspect of cluster analysis is computing distance between two clusters, so now we are going to start our discussion on this particular aspect, so how do we define a clusters, so this is quite a straight forward, the cluster is a set of one or more observations, right, so the cluster can have even a single observation as we have been discussing in previous lectures as well, so cluster is defined to have you know, cluster can be defined as a set of one or more observations.

Cluster Analysis

- Distance between two clusters
 - A cluster is a set of one or more observations
 - Distance metrics
 - Minimum Distance (Single Linkage), Maximum Distance (Complete Linkage), Average Distance (Average Linkage), and Centroid Distance

Cluster A: m observations

$A_1, A_2, A_3, \dots, A_m$

Cluster B: n observations

$B_1, B_2, B_3, \dots, B_n$

D_{AB} is the distance between two clusters

Now distance metrics that could be used to compute the distance between two clusters, so some of them are mentioned here, for example minimum distance metric is also called single linkage metrics, then we have maximum distance metric also called complete linkage, then we have average distance metric also called average linkage, then we have centroid distance. So let's discuss this metrics one by one, so let's consider two cluster, first one cluster A having M observations, A_1, A_2, A_3 up to A_M , and then the second cluster is cluster B having N observation, having you know observation B_1, B_2, B_3 up to B_N , and the distance between these two clusters, cluster A and cluster B can be defined as capital D_{AB} , so this is the distance between two clusters, so we are going to use these notations for our you know coming discussion.

Cluster Analysis

- Minimum Distance (Single Linkage)

- Distance between the pair of observations which are closest

$$D_{AB} = \min_{i=1,2,\dots,m; j=1,2,\dots,n} d_{ij}$$

- Maximum Distance (Complete Linkage)

- Distance between the pair of observations which are farthest

$$D_{AB} = \max_{i=1,2,\dots,m; j=1,2,\dots,n} d_{ij}$$



So let's you know discuss this first metric, minimum distance metric, single linkage, so how this can be defined? So distance between the pair of observations which are closest, so these two clusters, cluster A and B so we consider all the observation, all the observations you know between these clusters and identify the pair of observation, one observation belonging to cluster A, and the other observation belonging to cluster B which are closest, among all such pairs, now that particular you know, those particular observation and the distance between those particular observation is taken as the value, so which is defined as minimum distance metrics, so you can see the formula itself in the slide capital DAB, minimum of DIJ so all such DIJ, so DIJ could be where I is ranging from 1 to M, and J is ranging from 1 to N, so I is for denoting for all the observations that belong to cluster A, and J is denoting all the observation from 1 to N which belongs to cluster B, so you know we are supposed to find that particular pair of observations which are closest, and then that distance between those two observations is taken as the distance between clusters.

So this is the first metric for computation of distance between cluster, the second one is the maximum distance also called complete linkage, so how this particular metric is defined, so here we typically identify, typically find out the observations, you know, two observation, one belonging to cluster A, and the another one belonging to cluster B which are farthest from each other, so among all you know, among all pairs of observations one belonging to cluster 1, and the another belonging to the other cluster, and among all such pair of observations we identify, we find out the observations which are farthest from each other, and the distance between these two observations is taken as the distance between clusters, so the same you know ideas defined here in this slide, capital DAB where we are taking maximum of DIJ, where DIJ is indicating the distance between observation I and J, where I can be 1 to M, because it is referring to the cluster A and J could be 1 to N because it is referring to cluster B, so in this fashion the second metric, maximum distance metric defines the distance between 2 clusters.



Let's move forward, so our next metric is average distance or average linkage you know metric, so here what we do is, we take average of all possible distances between observations from 1 cluster to the other, so you know observations from 1 cluster, so one particular observation from

cluster A, and the other observation from cluster B and we take the distance, and similarly we identify all such pairs, you know, between observation of cluster A, and observation of cluster B, so in a sense it will include all you know, all the distances, all the observations, you know connecting to all the observation connecting to all the, of A, connecting to all the observation of cluster B, so all those distances are computed and the average of those distances is then taken as the you know average distance, so the same thing is defined here in the slide as well to capital DAB, average of DIJ, so DIJ is the distance between two observation, one belonging to cluster A, and the other one belonging to cluster B where I is referring to 1 to M, and J is referring to 1 to N, so here all possible distances between observations from cluster A to cluster B are then considered, and the average is computed and that average value is taken as the distance between two clusters.

Cluster Analysis

- **Average Distance (Average Linkage)**
 - Average of all possible distances between observations form one cluster to the other
$$D_{AB} = \text{average}(d_{ij})$$

where $i = 1, 2, \dots, m; j = 1, 2, \dots, n$
- **Centroid Distance**
 - Distance between two cluster centroids
 - Centroid is the vector of coordinates values computed by averaging measurements of each variable for all the observations in the cluster



22

Now this brings to our next metric that is centroid distance, so let's understand how this particular metric is defined, so centroid distance is defined as the distance between two clusters centroids, now when we say centroid we need to define this, so how centroid is define, centroid is the vector of coordinates values computed by averaging measurements of each variable for all the observations in the cluster, so for each cluster, for each variable and for each of those clusters we need to take average values within the cluster for that variable and that value will become one of the you know coordinate and similarly for other remaining variables will also compute this average values within the cluster for all the observation in a cluster, and that value, those value will become part of the coordinate for the centroid, and once these centroid and the coordinates of the centroid for each of those clusters are known to us, then we can compute the distance between those two centroids, so if we look at this centroid of cluster A

Cluster Analysis

- Centroid Distance

- Centroid for cluster A can be defined as

$$\bar{x}_A = \left(\frac{1}{m} \sum_{i=1}^m x_{1i}, \dots, \frac{1}{m} \sum_{i=1}^m x_{pi} \right)$$

- Distance between clusters A and B can be defined as

$$D_{AB} = d_{\bar{x}_A \bar{x}_B}$$



can be defined as below, so you can see \bar{x}_A , so we are referring to centroid A, so you can see the, you know here the M coordinates that are there, so first one, first coordinate is being computed as the average value so because there are, there are going to be, because there are M observations in cluster A so we have multiplied by $1/M$, and then summation for you know all the observation x_{1i} 's where 1 is representing the variable 1, and x_{pi} where i is ranging from 1 to M which is covering all the observations.

So we are taking the average of you know we are taking the average value for all the observation for a particular dimension that is variable 1, similarly for variable 2 and up to variable P, so because this is going to be a P dimensional vector, so \bar{x}_A which is the centroid for cluster A, this is a P dimensional vector and each of this P dimensions are being you know the value of, for each of this coordinates you know belonging to each of those P dimension are being computed by taking average value across all the observations, similarly we can compute for cluster B.

Now once the coordinates of you know centroid A and centroid B are known to us, we can define the distance between cluster A and cluster B as shown in the slide, so capital DAB as equal to distance between \bar{x}_A and \bar{x}_B , so \bar{x}_A and \bar{x}_B being the you know two points, and we can compute the distance between these two, so this is how the centroid distance metric can be used to compute the distance between two clusters.

Now, till now what we have understood lets you know go through some exercise in R, and understand few of this concepts, so let's go back to our R studio, so we'll have to repeat few of the you know lines again, so let's load the library. So let's import this dataset, let's take the copy of this particular dataset and we are already familiar with the structure of this dataset and the variables that are there, 17 variables in the original dataset, so if we are interested at looking at the observations, for 6 observation values we can see here, and then like we did for the you

```

1 library(xlsx)
2
3 # BreakfastCereals.xlsx
4 fulldf=read.xlsx(file.choose(), 1, header = T)
5 fulldf=fulldf[, !apply(is.na(fulldf), 2, all)]
6 df=fulldf
7
8 str(df)
9 head(df)
10
11 # changing scales of measurements
12 for(i in 1:13) {
13   df[,i+3]=1000*df[,i+3]/df[,3]
14 }
15 df=df[,-c(1,3)]
16
124 (Top Level) :

```

Environment History Connections

Data

- df 35 obs. of 17 variables
- fulldf 35 obs. of 17 variables

Files Plots Packages Help Viewer

```

C:/Users/T T Cell/Desktop/MOOC July 2018/Dr. Gaurav Dixit/Session 2/
$

```

	0.10	0	0.009	4.2
6	NA	0	0.450	8.8

```

Customer.Rating
1 4.2
2 3.0
3 4.4
4 4.4
5 4.9
6 3.3
> |

```

know previous, in the previous lectures as well, they are going to change the scales, and then we are sub-setting this, sub-setting a new dataset, for us new data frame, so this is the remaining, so in this one we have just the 15 variables, then plotting for two variables customer rating and price, this part we have already done in previous lectures normalization this was also done.

```

39 "NormRating"=sd(df[,NormRating]),
40 "NormPrice"=sd(df[,NormPrice]),
41 check.names = F))
42
43 # Distance matrix for first five cereals
44 DM=dist(df1[c(1:5),c(4,5)], method = "euclidean", diag = T,
45         upper = F); DM
46 DM1=as.matrix(DM)
47 DM1[upper.tri(DM1, diag = F)]=NA
48 rownames(DM1)=df1$Product[1:5]
49 colnames(DM1)=df1$Product[1:5]
50 DM1
51 print(DM1, digits = 2, na.print = "")
52
53 # Example:
54
181 (Top Level) :

```

Environment History Connections

Data

- df 35 obs. of 15 variables
- fulldf 35 obs. of 17 variables

Values

i 13L

Files Plots Packages Help Viewer

```

C:/Users/T T Cell/Desktop/MOOC July 2018/Dr. Gaurav Dixit/Session 2/
$

```

	0.2666667	0	0.0240000	11.20000
6	NA	0	1.2000000	23.46667

```

Customer.Rating
1 4.2
2 3.0
3 4.4
4 4.4
5 4.9
6 3.3
> |

```

So all this has been done, so distance metrics so this part let's go through this part, distance metrics for first five cereals, so out of the 35 cereals that we have in this particular dataset, we can use this particular function you know dist function to compute the distances between you know these 5 cereals, this is the function dist, if you are interested in finding more information about this function you can always go into the help section and type dist, and then you'll get

```

39   "NormRating"=sd(df1$NormRating),
40   "NormPrice"=sd(df1$NormPrice),
41   check.names = F))
42
43 # Distance matrix for first five cereals
44 DM=dist(df1[c(1:5),c(4,5)], method = "euclidean", diag = T,
45         upper = F); DM
46 DM1=as.matrix(DM)
47 DM1[upper.tri(DM1, diag = F)]=NA
48 rownames(DM1)=df1$Product[1:5]
49 colnames(DM1)=df1$Product[1:5]
50 DM1
51 print(DM1, digits = 2, na.print = "")
52
53 # Example:
54 <-----
181 (Top Level) >

```

```

C:/Users/T T Cell/Desktop/MOOC July 2018/Dr. Gaurav Dixit/Session 2/ >
5  0.2666667      0 0.0240000 11.20000
6  NA          0 1.2000000 23.46667

```

```

Customer.Rating
1      4.2
2      3.0
3      4.4
4      4.4
5      4.9
6      3.3
> |

```

The screenshot shows the RStudio interface. The source editor contains R code for computing a distance matrix. The console shows the output of the `dist` function and the resulting distance matrix. The environment pane on the right shows the objects `df`, `fulldf`, and `values`.

more information about this particular function what it does, and different arguments and the written values, so in the first argument as you can see here, we are passing on the first five rows and just you know two columns of it which are representing you know two variables that we are interested in, so these variables are part of DF1 which we need to compute before we move ahead, so let's compute DF1, so in the DF1 as you can see we are considering just two variables so one is customer rating, the another one is price, and then we have also computed the normalized value of these two variables, rating and price, so once this has been computed normalized value have been computed, now you can see here in the `dist` function the very first argument we are you know, we have selected 4 and 5 which are actually you know representing

```

30   check.names = F,,
37   data.frame("Standard Deviation", "Rating"=sd(df1$Rating),
38             "Price"=sd(df1$Price),
39             "NormRating"=sd(df1$NormRating),
40             "NormPrice"=sd(df1$NormPrice),
41             check.names = F))
42
43 # Distance matrix for first five cereals
44 DM=dist(df1[c(1:5),c(4,5)], method = "euclidean", diag = T,
45         upper = F); DM
46 DM1=as.matrix(DM)
47 DM1[upper.tri(DM1, diag = F)]=NA
48 rownames(DM1)=df1$Product[1:5]
49 colnames(DM1)=df1$Product[1:5]
50 DM1
51 <-----
321 (Top Level) >

```

```

C:/Users/T T Cell/Desktop/MOOC July 2018/Dr. Gaurav Dixit/Session 2/ >
27 -0.37165691
28 -0.95377356
29 -0.62902793
30  0.36811634
31  0.36811634
32  1.12271201
33  0.63761480
34  0.47591573
35 -0.44037901
> |

```

The screenshot shows the RStudio interface. The source editor contains R code for creating a data frame and computing a distance matrix. The console shows the output of the `dist` function. The environment pane on the right shows the objects `df`, `df1`, and `fulldf`.

the normalized values for rating and price, so only these two variables are being considered in the distance metric computation, so the first five cereals the distance between these you know first five cereals is being computed using normalized values of just two variables that is rating and price.

The screenshot displays the RStudio interface. The main editor shows R code for creating a data frame with normalized variables and computing a distance matrix. The console shows the execution of this code, resulting in a 5x5 distance matrix. A documentation window for the `dist` function is also visible.

```

30 check.names = F),
37   data.frame("Standard Deviation" = sd(df1$Price),
38             "Price" = sd(df1$Price),
39             "NormRating" = sd(df1$NormRating),
40             "NormPrice" = sd(df1$NormPrice),
41             check.names = F))
42
43 # Distance matrix for first five cereals
44 DM = dist(df1[c(1:5), c(4,5)], method = "euclidean", diag = T,
45          upper = F); DM
46 DM1 = as.matrix(DM)
47 DM1[upper.tri(DM1, diag = F)] = NA
48 rownames(DM1) = df1$Product[1:5]
49 colnames(DM1) = df1$Product[1:5]
50 DM1
51
461 (Top Level)

```

```

C:/Users/T T Cell/Desktop/MOOC July 2018/Dr. Gaurav Dixit/Session 2/
> # Distance matrix for first five cereals
> DM = dist(df1[c(1:5), c(4,5)], method = "euclidean", diag = T,
+         upper = F); DM
+
1 0.00000000
2 1.52509833 0.00000000
3 0.67468823 1.88638748 0.00000000
4 0.63815697 1.87359806 0.03962557 0.00000000
5 1.07330840 2.48866102 0.63593070 0.63563647 0.00000000
>

```

Distance Matrix Computation

Description

This function computes and returns the distance matrix computed by using the specified distance measure to compute the distances between the rows of a data matrix.

Usage

```
dist(x, method = "euclidean", diag = FALSE, ...)
```

You can see the second argument the method Euclidean so we are going to use Euclidean distance metric here, the other arguments are for the presentation of the result, so let's run this code, you can see in the output that the distance between the first five cereals have been computed, so this is in the metrics format so you know first cereal, distance of first cereal from itself is going to be 0 then distance of second cereal from the first one is this much 1.52, so these values we are able to see the values in this range because we are using normalized the scale, add in this function this metrics is essentially giving us information about distance between you know one cereal to another cereal this is just for the five cereals.

```

40 "NormPrice"=sd(df1$NormPrice),
41 check.names = F))
42
43 # Distance matrix for first five cereals
44 DM=dist(df1[c(1:5),c(4,5)], method = "euclidean", diag = T,
45 upper = F); DM
46 DM1=as.matrix(DM)
47 DM1[upper.tri(DM1, diag = F)]=NA
48 rownames(DM1)=df1$Product[1:5]
49 colnames(DM1)=df1$Product[1:5]
50 DM1
51 print(DM1, digits = 2, na.print = "")
52
53 # Example:
54 # First two cereals as Cluster A:
55 <
511 (Top Level) >

```

```

C:/Users/E T Cell/Desktop/MOOC July 2018/Dr. Gaurav Dixit/Session 2f >
> colnames(DM1)=df1$Product[1:5]
> DM1

```

	High Protein Cereal	Healthkart breakfast cereal	
High Protein Cereal	0.000000		NA
Healthkart breakfast cereal	1.5250983	0.000000	
Special K Multigrain and Honey	0.6746882	1.886387	
Chocos chocolate	0.6381570	1.873598	
Chocos Duet	1.0733084	2.488661	
	Special K Multigrain and Honey	Chocos chocolate	
High Protein Cereal		NA	NA
Healthkart breakfast cereal		NA	NA

If you're interested in changing the row and column names for these cereals to indicate the name of the cereals, we'll have to you know use, you know these particular code so we're going to run this, first create a metrics of this data and then few more you know transformation for example our triangular you know part of it, part of this metrics, we are making as NA , then we are changing the row names and column names, and then this is the metrics now, and because we have change the name of columns and rows, so it will take more space for display, now you can see the names of cereals, high protein cereal, health kart breakfast cereals and similarly others, these are being shown as the row names and similarly for column names, so each cereal and its distance from some other cereal is being shown in this metrics, and how these distances have been computed considering the normalized values of just two variables that is rating and price.

Now if you look at the values, these values are you know, number of decimal places have been used to accurately depict these values, distance values, if you're interested at looking at just you know the small metrics having just you know values up to two decimal point, we can run this part of code, now you can see the values have been rounded off, and we can see the distances between different cereals.

The screenshot displays the RStudio interface. The main editor shows R code for computing a distance matrix (DM1) between two clusters of cereals. The code includes comments explaining the clustering strategy: Cluster A contains 'High Protein Cereal' and 'Healthkart breakfast cereal', while Cluster B contains 'Special K Multigrain and Honey', 'Chocos Chocolate', and 'Chocos Duet'. The code uses the `dist` function with `method = "single"` to compute the distance matrix.

```

48 rownames(DM1)=df1$Product[1:5]
49 colnames(DM1)=df1$Product[1:5]
50 DM1
51 print(DM1, digits = 2, na.print = "")
52
53 # Example:
54 # First two cereals as Cluster A:
55 # High Protein Cereal, Healthkart breakfast cereal
56 # Next three cereals as Cluster B:
57 # Special K Multigrain and Honey, Chocos Chocolate, Chocos Duet
58
59 # Distance matrix with all possible distances
60 # between observations from one cluster to other
61 DM1[3:5, 1:2]
62 # Single linkage
63 <
531 (Top Level) >

```

The console output shows the resulting distance matrix for the first two clusters (rows 1 and 2) and columns 1 and 2:

```

Chocos Duet
> print(DM1, digits = 2, na.print = "")
      High Protein Cereal Healthkart breakfast cereal
High Protein Cereal           0.00
Healthkart breakfast cereal    1.53
Special K Multigrain and Honey  0.67
Chocos chocolate              0.64
Chocos Duet                   1.07

```

The right-hand pane shows the 'Distance Matrix Computation' documentation for the `dist` function, including a description and usage information.

Now since we have done our you know discussion on the distance between two clusters, we talked about single linkage, complete linkage and average linkage, let's see how this can be implemented in R, so if we you know, if we take this example where we take the first two cereals as cluster A, that means high protein cereal and health kart breakfast cereal, if we consider them as part of cluster A and the next three cereals that means special K multigrain and honey, and chocos, you know chocolate and chocos duet, you know these three cereals if we consider as part of cluster B as you can see here in the commented part of the code, then you know distance so now we have two cluster, cluster A and cluster B, and how distances between you know distance between these two clusters can be computed using different distance metric like single linkage, maximum linkage and you know maximum distance that is complete linkage and the average linkage, so let's see this.

The screenshot shows an R script in RStudio with the following code:

```

54 # FIRST two cereals as Cluster A:
55 #   High Protein Cereal, Healthkart breakfast cereal
56 # Next three cereals as Cluster B:
57 #   Special K Multigrain and Honey, chocos chocolate, Chocos Duet
58
59 # Distance matrix with all possible distances
60 # between observations from one cluster to other
61 DMI[3:5, 1:2]
62 # Single linkage
63 min(DMI[3:5, 1:2])
64 # Complete linkage
65 max(DMI[3:5, 1:2])
66 # Average linkage
67 mean(DMI[3:5, 1:2])
68
69
70
71 (Top Level)

```

The console output shows the execution of the code and the resulting distance matrix:

```

Chocos chocolate
Chocos Duet
> # Distance matrix with all possible distances
> # between observations from one cluster to other
> DMI[3:5, 1:2]

```

	High Protein Cereal	Healthkart breakfast cereal
Special K Multigrain and Honey	0.6746882	1.886387
Chocos chocolate	0.6381570	1.873598
Chocos Duet	1.0733084	2.488661

The right-hand side of the screenshot shows the R Environment pane with the following data objects:

- df: 35 obs. of 15 variables
- df1: 35 obs. of 5 variables
- DMI: num [1:5, 1:5] 0 1.525 0.6
- FullDf: 35 obs. of 17 variables

The R Documentation pane shows the help for the `dist` function:

Distance Matrix Computation

Description

This function computes and returns the distance matrix computed by using the specified distance measure to compute the distances between the rows of a data matrix.

Usage

```
dist(x, method = "euclidean", diag = FALSE, ...)
```

So before we go ahead let's look at the you know distance metric for all possible distances between observations from one cluster to other, so right now we have just you know two cluster, cluster A and cluster B, cluster A having you know first two cereals, and cluster B having the next three cereals, so this particular code will actually give us a metrics which is you know, which would depict the distances between the observations from one cluster to the other, so let's run this code, so you can see here, in the column names actually depict the cluster A cereals, high protein cereal and health kart breakfast cereals, and the row names are actually depicting the, you know, cluster B cereals which is special K and then chocos and chocolate and chocos duet, so now we can also see the distances between the observations from the you know cluster B to cluster A and vice-versa.

Now these are the distances, distance values that we are interested in, you know the distance metric you know for you know cluster distance that we have discussed so far, good actually require only these many values.

The screenshot shows RStudio with the following code in the editor:

```

55 # First two cereals as cluster A:
56 # High Protein Cereal, Healthkart breakfast cereal
57 # Next three cereals as cluster B:
58 # Special K Multigrain and Honey, Chocos Chocolate, Chocos Duet
59 # Distance matrix with all possible distances
60 # between observations from one cluster to other
61 DMI[3:5, 1:2]
62 # Single linkage
63 min(DMI[3:5, 1:2])
64 # Complete linkage
65 max(DMI[3:5, 1:2])
66 # Average linkage
67 mean(DMI[3:5, 1:2])
68
69 # Centroids for clusters A and B
70
71 (Top Level)

```

The console output is as follows:

```

C:/Users/T T Cell/Desktop/MOOC July 2018/Dr. Gaurav Dixit/Session 2/
> # between observations from one cluster to other
> DMI[3:5, 1:2]
              High Protein Cereal Healthkart breakfast cereal
Special K Multigrain and Honey    0.6746882                1.886387
Chocos chocolate                 0.6381570                1.873598
Chocos Duet                      1.0733084                2.488661
> # Single linkage
> min(DMI[3:5, 1:2])
[1] 0.638157
>

```

The Environment pane on the right shows the following data objects:

- df: 35 obs. of 15 variables
- df1: 35 obs. of 5 variables
- DMI: num [1:5, 1:5] 0 1.525 0.6
- fulldf: 35 obs. of 17 variables

The Help pane shows the documentation for `dist` (stats), titled "Distance Matrix Computation".

So single linkage as we talked about this is minimum distance, so we need to identify you know, in all these you know values which one is the you know lowest, so which two observations are closest, so we can just call this main function and we'll get this value, so let's run this, you can see 0.638157, if we try to identify cereals associated with this you know minimum distance then this is the distance, and we can see here chocos, chocolate and high protein cereal, so these are the two cereals which are closest to each other, so the distance between cluster A and cluster B using single linkage can actually be, will actually be this value 0.638157 and the observations which are closest or choco chocolate from cluster B, and high protein cereal from cluster A.

The screenshot shows RStudio with the following code in the editor:

```

55 # First two cereals as cluster A:
56 # High Protein Cereal, Healthkart breakfast cereal
57 # Next three cereals as cluster B:
58 # Special K Multigrain and Honey, Chocos Chocolate, Chocos Duet
59 # Distance matrix with all possible distances
60 # between observations from one cluster to other
61 DMI[3:5, 1:2]
62 # Single linkage
63 min(DMI[3:5, 1:2])
64 # Complete linkage
65 max(DMI[3:5, 1:2])
66 # Average linkage
67 mean(DMI[3:5, 1:2])
68
69 # Centroids for clusters A and B
70
71 (Top Level)

```

The console output is as follows:

```

C:/Users/T T Cell/Desktop/MOOC July 2018/Dr. Gaurav Dixit/Session 2/
> # between observations from one cluster to other
> DMI[3:5, 1:2]
              High Protein Cereal Healthkart breakfast cereal
Special K Multigrain and Honey    0.6746882                1.886387
Chocos chocolate                 0.6381570                1.873598
Chocos Duet                      1.0733084                2.488661
> # Single linkage
> min(DMI[3:5, 1:2])
[1] 0.638157
> # Complete linkage
> max(DMI[3:5, 1:2])
[1] 2.488661
>

```

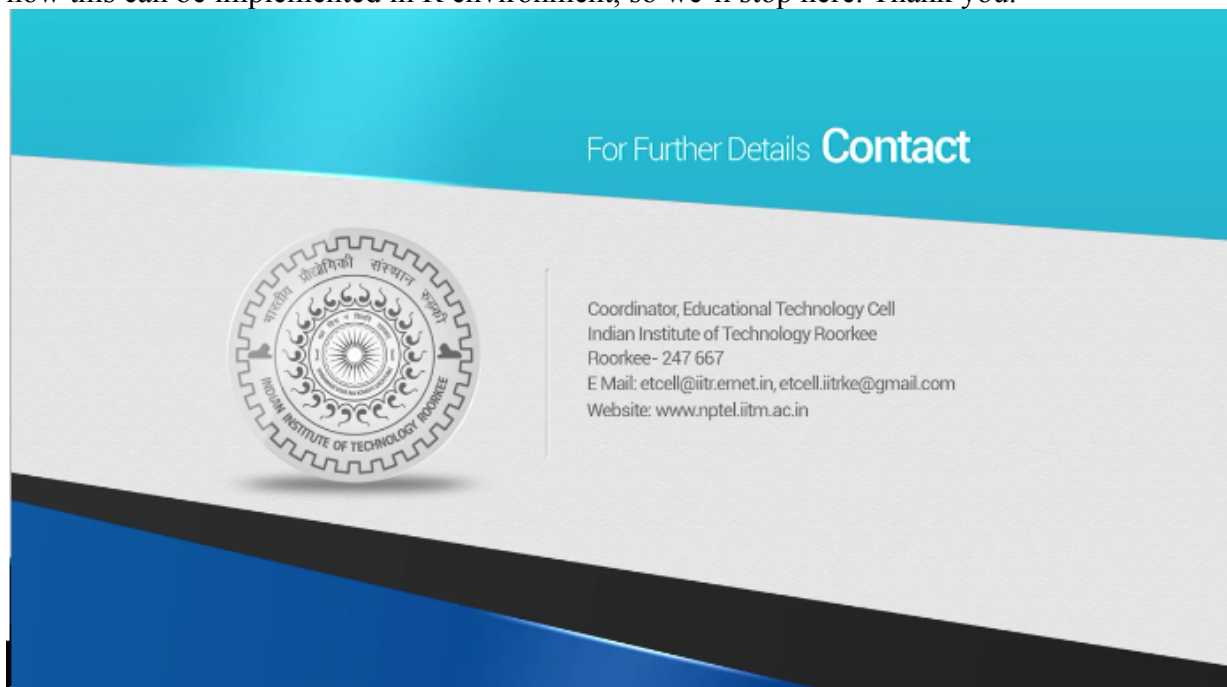
The Environment pane on the right shows the following data objects:

- df: 35 obs. of 15 variables
- df1: 35 obs. of 5 variables
- DMI: num [1:5, 1:5] 0 1.525 0.6
- fulldf: 35 obs. of 17 variables

The Help pane shows the documentation for `dist` (stats), titled "Distance Matrix Computation".

Similarly for the complete linkage distance metric we can just use this max function, because we know that it is the, we need to identify the two observation which are farthest from each other, so the max function will actually give us that value, so let's run this code as well, so we can see here the value comes out to be 2.488, you know this value and if we identify, find out this value in this metrics then we'll see that this is the value, and we see that this is the value and the cereals associated with this distance values are chocos duet and health kart breakfast cereal, so these are the two observation, chocos duet belonging to cluster B, and health kart breakfast cereal belonging to cluster A which are farthest from each other, and therefore this value becomes the you know distance between these two cluster using the complete linkage distance metric.

Similarly we can compute the distance between two clusters using the average linkage metrics, so we just need to compute the mean value for you know, all the possible distances are already you know depicted in this particular metrics, and we just need to compute the mean, so let's run this part of the code, you can see 1.43, so here all the you know distances that we just saw in this particular metrics, so they have been average of these values have been taken, and that becomes the distance between these two clusters, cluster A and cluster B, so in this fashion as we you know, so we have been able to discuss the different distance metrics that could be used to compute the distance between two clusters, and we also do an exercise in R where we saw how this can be implemented in R environment, so we'll stop here. Thank you.



For Further Details **Contact**

Coordinator, Educational Technology Cell
Indian Institute of Technology Roorkee
Roorkee - 247 667
E Mail: etcell@iitr.ernet.in, etcell.iitrke@gmail.com
Website: www.nptel.iitm.ac.in

For Further Details Contact
Coordinator Educational Technology Cell
Indian Institute of Technology Roorkee
Roorkee – 247 667
E Mail:-etcell@iitr.ernet.in, iitrke@gmail.com
Website: www.nptel.iitm.ac.in

Acknowledgement

Prof. Ajit Kumar Chaturvedi
Director, IIT Roorkee

NPTEL Coordinator

IIT Roorkee

Prof. B. K Gandhi

Subject Expert

Dr. Gaurav Dixit

Department of Management Studies

IIT Roorkee

Produced by

Mohan Raj.S

Graphics

Binoy V.P

Web Team

Dr. Nibedita Bisoyi

Neetesh Kumar

Jitender Kumar

Vivek Kumar

Dharamveer Singh

Gaurav Kumar

An educational Technology cell

IIT Roorkee Production

© Copyright All Rights Reserved

WANT TO SEE MORE LIKE THIS

SUBSCRIBE