Lab8 - Cluster Analysis with R

Clustering a set of objects into groups is usually moved by the aim of identifying internally homogenous groups according to a specific set of variables. In order to accomplish this objective, the starting point is computing a matrix, called *dissimilarity matrix*, which contains information about the dissimilarity of the observed units. According to the nature of the observed variables (quantitative, qualitative, binary or mixed type variables), we can define and use different measures of dissimilarity.

1. Repeat the following examples
2. Apply complete clustering analysis to Lab8\_priorities\_perifereion dataset

**Example 1**

Data Longley contained in the R-package AER consists of the number of people employed from 1947-1962.

> library(AER)

> data("Longley")

> longley<-as.data.frame(Longley)

> x<-longley$employment/1000

> x

[1] 60323 61122 60171 61187 63221 63639 64989 63761 66019 67857 68169 [12] 66513 68655 69564 69331 70551

> label.x<-as.character(c(1947:1962))

Calculate the dissimilarity between observations using the Euclidean distance

> dist.l<-dist(x,method="euclidean")

Compute a hierarchical cluster analysis on the distance matrix using the complete linkage method

> h<-hclust(dist.l, method="complete")

> print(h)

Call:

hclust(d = dist.l, method = "complete")

Cluster method : complete Distance : euclidean Number of objects: 16

In order to see all the steps of the clustering type:

> h$merge

[,1] [,2] [1,] -2 -4 [2,] -6 -8

|  |  |  |
| --- | --- | --- |
| [3,] | -1 | -3 |
| [4,] | -14 | -15 |
| [5,] | -10 | -11 |
| [6,] | -9 | -12 |
| [7,] | -5 | 2 |
| [8,] | -13 | 5 |
| [9,] | 1 | 3 |
| [10,] | -16 | 4 |
| [11,] | -7 | 6 |
| [12,] | 8 | 10 |
| [13,] | 7 | 11 |
| [14,] | 9 | 13 |
| [15,] | 12 | 14 |

Note that the minus in front of the unit number indicates that this is a single observation being merged;

whereas numbers alone indicate the step at which the considered clusters were built. Create a plot of the clustering tree

> plclust(h,labels=label.x)

> title("Dendrogram of employment figures: Complete linkage")

**Dendrogram of employment figures: Complete linkage**

Height

0

2

4

6

8

10

1959

1956

1957

1962

1960

1961

1948

1950

1947

1949

1951

1952

1954

1953

1955

1958

dist.l

hclust (\*, "complete")

What is an appropriate number of clusters according to this plot?

A common choice is to cut the tree by the largest difference of heights between two nodes. The height values are contained in the output of hclust function:

> h.cl<-h$height # height values

> h.cl

[1] 0.065 0.122 0.152 0.233 0.312 0.494 0.540 0.798 1.016

1.220 1.524 2.694 3.292 6.342 10.380

> h.cl2<-c(0,h.cl[-length(h.cl)]) # vector that has to be substracted

+ from h.cl

> round(h.cl-h.cl2,3) # differences in height, rounded at the 3rd digit

[1] 0.065 0.057 0.030 0.081 0.079 0.182 0.046 0.258 0.218 0.204 0.304

1.170 0.598 3.050 4.038

> max(round(h.cl-h.cl2,3)) # the largest increase

[1] 4.038

> which.max(round(h.cl-h.cl2,3)) # the step of the largest increase

[1] 15

According to this approach, the appropriate number of cluster is two, because the largest difference is at the last step of the merging process.

Compute a hierarchical cluster analysis on the distance matrix using the average linkage method:

> h<-hclust(dist.l,method="average")

> print(h)

Call:

hclust(d = dist.l, method = "average")

Cluster method : average Distance : euclidean Number of objects: 16

> plclust(h,labels=label.x)

> title("Dendrogram of employment figures: Average linkage")

In order to choose where to cut the three, the differences in the height values are evaluated:

> h.cl<-h$height # height values

> h.cl

[1] 0.065000 0.122000 0.152000 0.233000 0.312000 0.479000 0.494000

0.642000 0.907500 1.103500 1.277000 1.588333 2.300000 3.989583

5.926667

> h.cl2<-c(0,h.cl[-length(h.cl)]) # vector that has to be substracted

+ from h.cl

> round(h.cl-h.cl2,3) # differences in height, rounded at the 3rd digit

[1] 0.065 0.057 0.030 0.081 0.079 0.167 0.015 0.148 0.265 0.196 0.173

0.311 0.712 1.690 1.937

> max(round(h.cl-h.cl2,3)) # the largest increase

[1] 1.937

> which.max(round(h.cl-h.cl2,3)) # the step of the largest increase

[1] 15

Again, the number of clusters that seems more appropriate is two, since the largest increase in the height values is at the last step of the merging process.

**Dendrogram of employment figures: Average linkage**

Height

2

3

4

5

6

1957

1961

1947

1951

1952

1954

1953

1955

1958

dist.l

1950

hclust (\*, "average")

1956

1960

**Example 2**

Consider data set lifeexp.dat. It contains life expectancy in the 1960s distinguished by country, age and sex.

Perform single, complete and average linkage agglomerative hierarchical cluster based on the Euclidean distance measure and produce suitable plots.

life<-read.table("c:\\temp\\lifeexp.dat",header=TRUE)

rownames(life)<-life[,1]

country <-rownames(life)

par(mfrow=c(1,3))

India

Pakistan

25

plclust(hclust(dist(life), method="single"), labels=country , ylab="Distance")

Mexico

Romania

Argentina Ukraine Turkey KoreaSouth China KoreaNorth

Brazil Vietnam Philippines Peru

Iran

Morocco

Indonesia

Kenya

Ethiopia

title("Single Linkage")

plclust(hclust(dist(life), method="complete"), labels=country, ylab="Distance")

SouthAfrica

5

title("Complete Linkage")

0

plclust(hclust(dist(life), method="average"), labels=country , ylab="Distance")

UnitedKingdom

Distance

0

5

10

Poland

Mexico

Romania

Russia

Thailand

Colombia

Argentina Ukraine Turkey KoreaSouth China KoreaNorth

Japan France Italy

Spain

Canada

Germany

UnitedKingdom

Venezuela

Taiwan UnitedStates Egypt

Indonesia Kenya SouthAfrica

Brazil Vietnam Philippines Peru

Iran

Morocco

India

Pakistan

Ethiopia

Myanmar(Burma)

Zaire

Tanzania

Bangladesh

Sudan

title("Average Linkage")

3.0

3.5

20

**Single Linkage**

Distance

0.0

0.5

1.0

1.5

2.0

2.5

Japan

France

Italy

Spain

Venezuela

Taiwan UnitedStates Canada

Russia

Thailand

Poland

Colombia

Egypt

Myanmar(Burma) Zaire Tanzania Bangladesh Sudan

Distance

10

15

**Complete Linkage**

**Average Linkage**

dist(lif e)

hclust (\*, "single")

dist(lif e)

hclust (\*, "complete")

Germany

Poland

Mexico

Romania

Russia Thailand Colombia

Argentina Ukraine Turkey KoreaSouth China KoreaNorth

Japan

France

Italy Spain Canada

Germany

UnitedKingdom

Venezuela

Taiwan UnitedStates Egypt

Indonesia

Kenya SouthAfrica Brazil

Vietnam Philippines Peru

Iran

Morocco

India

Pakistan

Myanmar(Burma) Zaire

Ethiopia

Tanzania

Bangladesh

Sudan

dist(lif e)

hclust (\*, "average")

The singe linkage method shows a high degree of asymmetry – an example of 'chaining', i.e. result of

'prematurely' combining individuals/clusters using the minimum distance criteria which defines single linkage.

The complete linkage method is much more balanced in the way it forms clusters, producing four or five clear clusters.

The average linkage method returns clusters which again show asymmetry, but not as pronounced as single linkage.

Suppose to cut the complete linkage dendrogram at height 8: it would yield 4 clusters. Try to construct four clusters using the *k*-means algorithm as follows:

clusters.km <- kmeans(life,4) country.clus.km <- lapply(1:4,function(nc) country[clusters.km$cluster==nc])

> country.clus.km

[[1]]

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| [1] | "Canada" | "France" | "Germany" | "Italy" |
| [5] | "Japan" | "Spain" | "Taiwan" | "UnitedKingdom" |
| [9] | "UnitedStates" | "Venezuela" |  |  |

[[2]]

|  |  |  |  |
| --- | --- | --- | --- |
| [1] "Brazil" | "Egypt" | "Indonesia" | "Iran" "Kenya" |
| [6] "Morocco" | "Peru" | "Philippines" | "SouthAfrica" "Vietnam" |

|  |  |  |  |
| --- | --- | --- | --- |
| [[3]] |  | | |
| [1] "Bangladesh" | "Ethiopia" | "India" | "Myanmar(Burma)" |
| [5] "Pakistan" | "Sudan" | "Tanzania" | "Zaire" |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| [[4]] |  | | | |
| [1] | "Argentina" | "China" | "Colombia" | "KoreaNorth" "KoreaSouth" |
| [6] | "Mexico" | "Poland" | "Romania" | "Russia" "Thailand" |
| [11] | "Turkey" | "Ukraine" |  |  |

Do these appear to be sensible groupings?

From our limited knowledge of demographic properties, the groupings appear consistent, eg. developing countries grouped together, developed countries grouped together.

**Example 3**

Consider again the sparrowElphick dataset. Now try to use the k-means cluster analysis to check whether there are two clusters, one containing the sparrows which survived and the other the sparrows which died.

Using the R-function kmeans(), select two clusters and obtain the following output:

> sparrows<-read.table("…/sparrowsElphic.dat",header=T)

> kmeans(sparrows, 2)

K-means clustering with 2 clusters of sizes 24, 25k

Cluster means:

totL AlarE bhL hL kL

1 160.9167 245.4583 31.90417 18.80833 21.29583

2 155.1600 237.3600 31.03200 18.14400 20.37600

Clustering vector:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| [1] 1 | 2 2 | 2 2 | 1 2 | 2 1 2 2 1 | 1 1 | 2 2 1 2 2 1 2 | 2 2 1 2 1 2 | 1 1 2 1 1 1 1 |
| [35] 2 | 1 2 | 1 2 | 1 1 | 2 2 1 2 1 | 2 1 | 1 |  |  |

Within cluster sum of squares by cluster: [1] 361.0892 371.5216

(between\_SS / total\_SS = 62.7 %)

Cluster sizes: [1] 25 24

Available components:

|  |  |  |
| --- | --- | --- |
| [1] "cluster" "centers" | "totss" | "withinss" |
| [5] "tot.withinss" "betweenss" | "size" |  |

Do the two clusters contain dead/alive sparrows respectively? No.

Run now the k-means clustering allowing the number of clusters to vary from 2 to 5 and compare the results in terms of ASW and PG indexes.

ASW and PG indexes can be obtained with the function cluster.stats, in package fpc.

> library(fpc)

> ?cluster.stats

> dist.data<-dist(sparrows,"euclidean")

> cl2<-kmeans(sparrows, 2)

> out.cl2<-cluster.stats(dist.data,cl2$cluster)

> out.cl2$avg.silwidth

[1] 0.5046398

> out.cl2$pearsongamma

[1] 0.6496322

> cl3<-kmeans(sparrows, 3)

> out.cl3<-cluster.stats(dist.data,cl3$cluster)

> out.cl3$avg.silwidth

[1] 0.3835227

> out.cl3$pearsongamma

[1] 0.5833999

> cl4<-kmeans(sparrows, 4)

> out.cl4<-cluster.stats(dist.data,cl4$cluster)

> out.cl4$avg.silwidth

[1] 0.3427861

> out.cl4$pearsongamma

[1] 0.5726651

> cl5<-kmeans(sparrows, 5)

> out.cl5<-cluster.stats(dist.data,cl5$cluster)

> out.cl5$avg.silwidth

[1] 0.3667793

> out.cl5$pearsongamma

[1] 0.5626833

According to the values of the two indexes, the k-means with two clusters produces more homogenous groups; the second best choice would be fixing the number *k* of clusters equal to 3.

A meaningful representation of the clustering outcome is the so called 'silhouette plot'. On the x-axis it shows the silhouette width for each observation in the corresponding cluster; units in the same cluster are plotted in decreasing order according to their silhouette value. Different clusters are separately plotted. Furthermore, it reports the number of observations in each cluster and the average silhouette width of the classification.

In order to produce a silhouette plot we need to load the cluster library and to use function

silhouette:

> library(cluster)

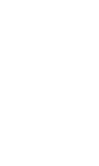
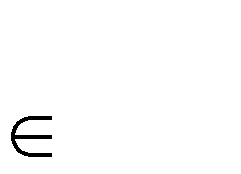
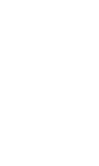
> sil<-silhouette(cl2$cluster,dist.data)

> plot(sil,cex.names=0.6,nmax=98,main="Silhouette Plot for 'Sparrows'

+ dataset, k=2")

**Silhouette Plot for 'Sparrows' dataset, k=2**

n = 49 2clusters Cj



j : nj | avei Cj si

27

19

4

16

45

47

42

39

18

15

25

7

30 1 : 25 | 0.52

8

2

3

37

22

23

21

10

43

11

5

35

6

46

20

13

48

49

9

44

40

26

29

36 2 : 24 | 0.49

33

12

32

34

41

17

28

14

24

1

38

31

0.0 0.2 0.4 0.6 0.8 1.0

Silhouette width si

Average silhouette width : 0.5

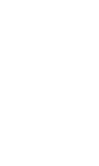
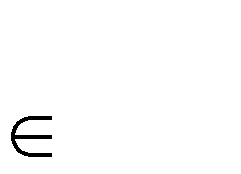
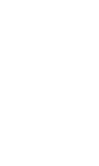
Or, with coloured clusters:

> plot(sil,cex.names=0.6,nmax=98,main="Silhouette Plot for 'Sparrows'

+ dataset, k=2", col=c("red","green"))

**Silhouette Plot for 'Sparrows' dataset, k=2**

n = 49 2clusters Cj



j : nj | avei Cj si

27

19

4

16

45

47

42

39

18

15

25

7

30 1 : 25 | 0.52

8

2

3

37

22

23

21

10

43

11

5

35

6

46

20

13

48

49

9

44

40

26

29

36 2 : 24 | 0.49

33

12

32

34

41

17

28

14

24

1

38

31

0.0 0.2 0.4 0.6 0.8 1.0

Silhouette width si

Average silhouette width : 0.5