Statistical Inference

Lab 4

Kmeans

Clustering

- It is basically a type of ***unsupervised learning method***.
- An unsupervised learning method is a method in which we draw references from datasets consisting of input data ***without labeled responses***.

- ***Clustering*** is the task of dividing the population or data points into a number of groups such that data points in the same groups are more similar to other data points in the same group and dissimilar to the data points in other groups.

K-means

- The K-Means function, provided by the cluster package, is used as follows:

```
install.packages("cluster")        # Installing Packages

library(cluster)                   # Loading package
```

> **kmeans** (x, centers, iter.max = 10, nstart = 1, algorithm = c("Hartigan-Wong", "Lloyd", "Forgy", "MacQueen"))

- ***where the arguments are:***

- **x:** A numeric matrix of data, or an object that can be coerced to such a matrix
(such as a numeric vector or a data frame with all numeric columns).

- **centers:** Either the number of clusters or a set of initial (distinct) cluster centers.
If a number, a random set of (distinct) rows in x is chosen as the initial centers.

- **iter.max:** The maximum number of iterations allowed.

- **nstart:** If centers is a number, nstart gives the number of random sets that
should be chosen.

- **algorithm:** The algorithm to be used. It should be one of the following "Hartigan-
Wong", "Lloyd", "Forgy" or "MacQueen". If no algorithm is specified, the
algorithm of Hartigan and Wong is used by default.

Output

kmeans returns an object of class "kmeans" .It is a list with at least the following components:

cluster

A vector of integers (from 1:k) indicating the cluster to which each point is allocated.

centers

A matrix of cluster centres.

totss

The total sum of squares.

withinss

Vector of within-cluster sum of squares, one component per cluster.

tot.withinss

Total within-cluster sum of squares, i.e.sum(withinss).

betweenss

The between-cluster sum of squares, i.e.totss-tot.withinss.

size

The number of points in each cluster.

The Iris Dataset

The data set consists of 50 samples from each of three species of *Iris* (*Iris setosa*, *Iris virginica* and *Iris versicolor*). Four features were measured from each sample: the length and the width of the sepals and petals, in centimetres. Based on the combination of these four features, a linear discriminant model used to distinguish the species from each other

	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
36	5.0	3.2	1.2	0.2	setosa
37	5.5	3.5	1.3	0.2	setosa
38	4.9	3.6	1.4	0.1	setosa
39	4.4	3.0	1.3	0.2	setosa
40	5.1	3.4	1.5	0.2	setosa
41	5.0	3.5	1.3	0.3	setosa
42	4.5	2.3	1.3	0.3	setosa
43	4.4	3.2	1.3	0.2	setosa
44	5.0	3.5	1.6	0.6	setosa
45	5.1	3.8	1.9	0.4	setosa
46	4.8	3.0	1.4	0.3	setosa
47	5.1	3.8	1.6	0.2	setosa
48	4.6	3.2	1.4	0.2	setosa
49	5.3	3.7	1.5	0.2	setosa
50	5.0	3.3	1.4	0.2	setosa
51	7.0	3.2	4.7	1.4	versicolor
52	6.4	3.2	4.5	1.5	versicolor
53	6.9	3.1	4.9	1.5	versicolor
[illegible]	5.5	2.3	4.0	1.3	versicolor
[illegible]	6.5	2.8	4.6	1.5	versicolor
[illegible]	5.7	2.8	4.5	1.3	versicolor

Iris Setosa

Iris Virginica

Iris Versicolor

example

- Explore the iris data set

```
> newiris=read.csv('path/iris.csv’)   #Read from excel
```

Or

```
>iris                                  #loaded in R
>newiris <- iris

>newiris$Species <- NULL               #Set column value to NULL
>newiris
```

- Apply kmeans to newiris, and store the clustering result in kc.

```
>kc <- kmeans(newiris, 3)
>kc
```

```
K-means clustering with 3 clusters of sizes 38, 50, 62

Cluster means:
  Sepal.Length Sepal.Width Petal.Length Petal.Width
1     6.850000    3.073684     5.742105    2.071053
2     5.006000    3.428000     1.462000    0.246000
3     5.901613    2.748387     4.393548    1.433871

Clustering vector:
  [1] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
 [51] 3 3 1 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 1 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3
[101] 1 3 1 1 1 1 3 1 1 1 1 1 1 3 3 1 1 1 1 3 1 3 1 3 1 1 3 3 1 1 1 1 1 3 1 1 1 1 3 1 1 1 3 1
[151] 1 1
```

- >kc$cluster

[1] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
[51] 3 3 1 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 1 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3
[101] 1 3 1 1 1 1 3 1 1 1 1 1 1 3 3 1 1 1 1 3 1 3 1 3 1 1 3 3 1 1 1 1 1 3 1 1 1 1 3 1 1 1 3 1 1 1 3 1 1 3

>kc$center

	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width
1	6.850000	3.073684	5.742105	2.071053
2	5.006000	3.428000	1.462000	0.246000
3	5.901613	2.748387	4.393548	1.433871

Compare the Species label with the clustering result

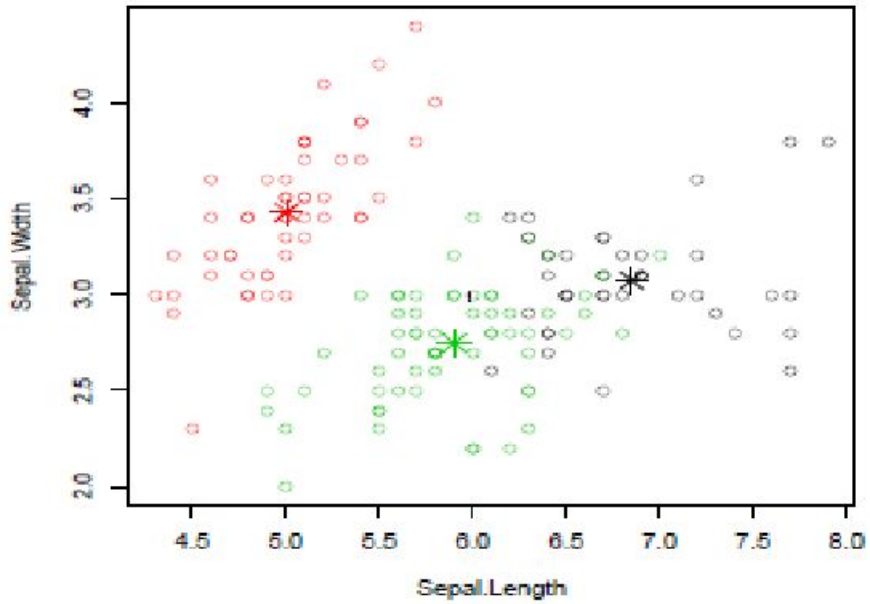
- >table(iris$Species, kc$cluster)

	1	2	3
setosa	0	50	0
versicolor	2	0	48
virginica	36	0	14

Plot the clusters and their centres.

- plot(newiris$Sepal.Length, newiris$Sepal.Width, col = kc$cluster)

> points(kc$centers, col = 1:3, pch = 8, cex=2)



To know the appropriate number of clusters

```
>wss=numeric(16)
>for(i in 1:16) wss[i]=sum(kmeans(newiris,i)$withinss)
>plot(1:16,wss,type='b',xlab='number of cluster',ylab='within grp sum square')
```


Thank you