**R: K-Means Clustering**

*Goal:* Perform K-means clustering to identify groups of similar observations

*Data:* Iris dataset from base R

**Table of Contents**

1 --- Data Description and Setup

2 --- K-means Clustering

**Data Description and Setup**

We will be using the Iris Dataset which, if you recall, includes the following variables describing flower observations:

* **Sepal.Length** - length of outer section of flower
* **Sepal.Width** - width of outer section of flower
* **Petal.Length** - length of flower's petal
* **Petal.Width** - width of flower's petal

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

* **Species** - type of Iris flower

(Note: We have 3 species or clusters which we already know; when iris data are used to test clustering methods, we assume we don’t know these specie labels a priori. That is, we only use **Sepal.Length**, **Sepal.Width, Petal.Length and Petal.Width for clustering**).

We begin by importing the data and necessary libraries.

library(ggplot2)

library(cluster) # for silhouette plot

set.seed(1)

data(iris)

head(iris)

Sepal.Length Sepal.Width Petal.Length Petal.Width Species

1 5.1 3.5 1.4 0.2 setosa

2 4.9 3.0 1.4 0.2 setosa

3 4.7 3.2 1.3 0.2 setosa

4 4.6 3.1 1.5 0.2 setosa

5 5.0 3.6 1.4 0.2 setosa

6 5.4 3.9 1.7 0.4 setosa

**K-means Clustering**

Given a predefined number of clusters, K-means clustering groups observations based on some distance metric, usually Euclidean distance. The algorithm iterative moves cluster centroids to optimize within-cluster sum of squares until some stopping criterion is reached. The **kmeans** function requires several arguments:

* **x**: a matrix or data frame containing the observations you want to cluster
* **centers**: the number of clusters to find
* **iter.max**: maximum number of iterations to allow K-means to run
* **nstart**: how many times to run the algorithm; multiple runs are often necessary since k-means may converge to a local minima, so we take the best result

To select an optimal number of clusters for K-means, an elbow plot and average silhouette score are usually used.

To create an elbow plot, we simply perform K-means clustering with different numbers of clusters, from 2 up to a specified maximum.

Rule of thumb: the max. number of clusters could be specified as the square root of half of the sample size, e.g., the sample size of Iris is 150, then about 10 clusters. Then, we plot the total within-cluster sum of squares of all clusters (y-axis) against the number of clusters (x-axis)

## Elbow Plot

results <- list()

tot.withinss <- numeric(9)

for(i in 2:10){

results[[i-1]] <- kmeans(iris[,1:4], centers=i, iter.max=20, nstart=5)

tot.withinss[i-1] <- results[[i-1]]$tot.withinss

}

plot(2:10, tot.withinss, type="b", pch=19, xlab="Number of Clusters", ylab = "Total Within-Cluster Sum of Squres")

This plot is called an “elbow plot” because the point at which the evaluation metric stops decreasing rapidly is called the “elbow” – this point seems to denote the optimal number of clusters as either 3 or 4. So, we need additional validation metric, e.g, average silhouette score, as well as visualization or mapping methods to help us identify the optimal number of clusters.

Chart, line chart

Description automatically generated

While within-cluster sum of squares is acceptable, another visual metric to evaluate the optimal number of clusters is **average silhouette score**. The silhouette score is a measure of how similar an object is to its own cluster (cohesion) compared to other clusters (separation).

e.g. The silhouette ranges from -1 to 1, where a high value indicates that the object is well matched to its own cluster and poorly matched to neighboring clusters.

* If most objects have a high value, then the clustering configuration is appropriate.
* If many points have a low or negative value, then the clustering configuration may have too many or too few clusters.

Below, we plot the silhouette across observations for 3 and 4 clusters, which also show how many observations occur in each cluster and the average silhouette score per cluster. These are followed by an elbow plot of the mean silhouette score for all clustering solutions.

## Silhouette Analysis

d <- dist(iris[,1:4]) # Calculate distance matrix

sils <- list()

avg.sil <- numeric(9)

for(i in 1:9){

output <- silhouette(results[[i]]$cluster, d)

# average the scores

sils[[i]] <- output

avg.sil[i] <- mean(output[,3])

}

# Obtain the silhouette scores for 3 and 4 clusters

sil3 <- sils[[2]]

sil4 <- sils[[3]]

# Plot the scores for each observation

plot(sil3, main="Silhouette Plot: KMeans, 3 Clusters", col=1:3, border=NA)

Chart

Description automatically generated

plot(sil4, main="Silhouette Plot: KMeans, 4 Clusters", col=1:4, border=NA)

Chart

Description automatically generated

# Plot the scores

plot(2:10, avg.sil, type="b", pch=19, xlab="Number of Clusters", ylab = "Average Silhouette Score")

Chart, line chart

Description automatically generated

We observe that most observations in each cluster have high silhouette scores, with a small minority of low-scoring outliers. Both have a single cluster with a much higher silhouette score than others. The average silhouette score levels off at 4 clusters.

Now let us plot where these clusters are in the data and determine how accurate they are to the true **Species** labels in the data:

# Add clusters to the data frame

iris$Cluster3 <- factor(results[[2]]$cluster)

iris$Cluster4 <- as.factor(results[[3]]$cluster)

# Plot clusters on two selected attributes

ggplot(iris) + geom\_point(aes(x=Sepal.Width, y=Petal.Length, color=Cluster3), size=3) + theme\_minimal()

Chart, scatter chart

Description automatically generated

ggplot(iris) + geom\_point(aes(x=Sepal.Width, y=Petal.Length, color=Cluster4), size=3) + theme\_minimal()

Chart, scatter chart

Description automatically generated

# 3 Cluster Metrics

results[[2]]$tot.withinss

[1] 78.85144

avg.sil[2]

[1] 0.552819

### Confusion Tables

# The true 3 clusters: setosa (n=50); versicolor (n=50); virginica (n =50)

# Cluster 3 corresponds to setosa

# Cluster 2 corresponds mostly to versicolor

# Cluster 1 corresponds mostly to virginica

iris$Cluster3 <- factor(iris$Cluster3, c(3,2,1))

conf3 <- table(iris$Cluster3, iris$Species)

conf3

setosa versicolor virginica

3 50 0 0

2 0 48 14

1 0 2 36

acc <- sum(diag(conf3)) / nrow(iris)

acc

[1] 0.8933333

# 4 Cluster Metrics

results[[3]]$tot.withinss

[1] 57.22847

avg.sil[3]

[1] 0.4980505

# The true 3 clusters: setosa (n=50); versicolor (n=50); virginica (n=50)

# Cluster 4 corresponds to setosa

# Cluster 3 corresponds mostly to versicolor

# Cluster 2 corresponds mostly to virginica

# Cluster 1 (n=40) captures additional observations from versicolor (23) and virginica (17)

iris$Cluster4 <- factor(iris$Cluster4, c(4,2,3,1))

conf4 <- table(iris$Cluster4, iris$Species)

conf4

setosa versicolor virginica

4 50 0 0

2 0 27 1

3 0 0 32

1 0 23 17

acc <- sum(diag(conf4)) / nrow(iris)

acc

[1] 0.7266667

The graphs provide visual confirmation that the clusters are coherent groups near each other in the data, and the confusion tables and accuracies further demonstrate that the clusters are similar to the actual groups of species.

The following table summarizes the clustering metrics across solutions with both 3 and 4 clusters:

|  |  |  |
| --- | --- | --- |
| # of Clusters | 3 | 4 |
| Tot. Within-Cluster SS | 78.85 | 57.22 |
| Mean Silhouette Score | 0.55 | 0.50 |
| Clustering Accuracy | 0.89 | 0.73 |