K-Means and K-Medoids Clustering on Simulated Data

Halil Bisgin

Introduction

In this analysis, we simulate data to apply both **K-Means** and **K-Medoids** clustering algorithms, compare their results, and visualize the clustering outcomes.

We also measure the time taken for each clustering method.

Load Required Libraries

```r
# Install libraries if not already installed
# install.packages("cluster")
# install.packages("factoextra")
# install.packages("ggplot2")
# install.packages("tictoc")

# Load necessary libraries
library(cluster)      # For K-Medoids (PAM)
library(factoextra)   # For visualization
library(ggplot2)      # For plotting
library(tictoc)       # For measuring execution time
```

```r
# Set random seed for reproducibility
set.seed(42)

# Create 3 clusters with normal distribution
n <- 1500 # Total number of data points
cluster1 <- data.frame(x = rnorm(n/3, mean = 2, sd = 0.5), y = rnorm(n/3, mean = 2, sd = 0.5))
cluster2 <- data.frame(x = rnorm(n/3, mean = 6, sd = 0.5), y = rnorm(n/3, mean = 6, sd = 0.5))
cluster3 <- data.frame(x = rnorm(n/3, mean = 10, sd = 0.5), y = rnorm(n/3, mean = 2, sd = 0.5))

# Combine clusters into one dataset
data <- rbind(cluster1, cluster2, cluster3)
colnames(data) <- c("feature1", "feature2")

# Normalize the data
scaled_data <- scale(data)
```

Apply K-Means Clustering

```
tic("K-Means")  # Start timer
kmeans_result <- kmeans(scaled_data, centers = 3, nstart = 10)
toc()  # Stop timer
```

```
## K-Means: 0.01 sec elapsed
```

Apply K-Medoids Clustering

```
tic("K-Medoids")  # Start timer
kmedoids_result <- pam(scaled_data, 3)
toc()  # Stop timer
```

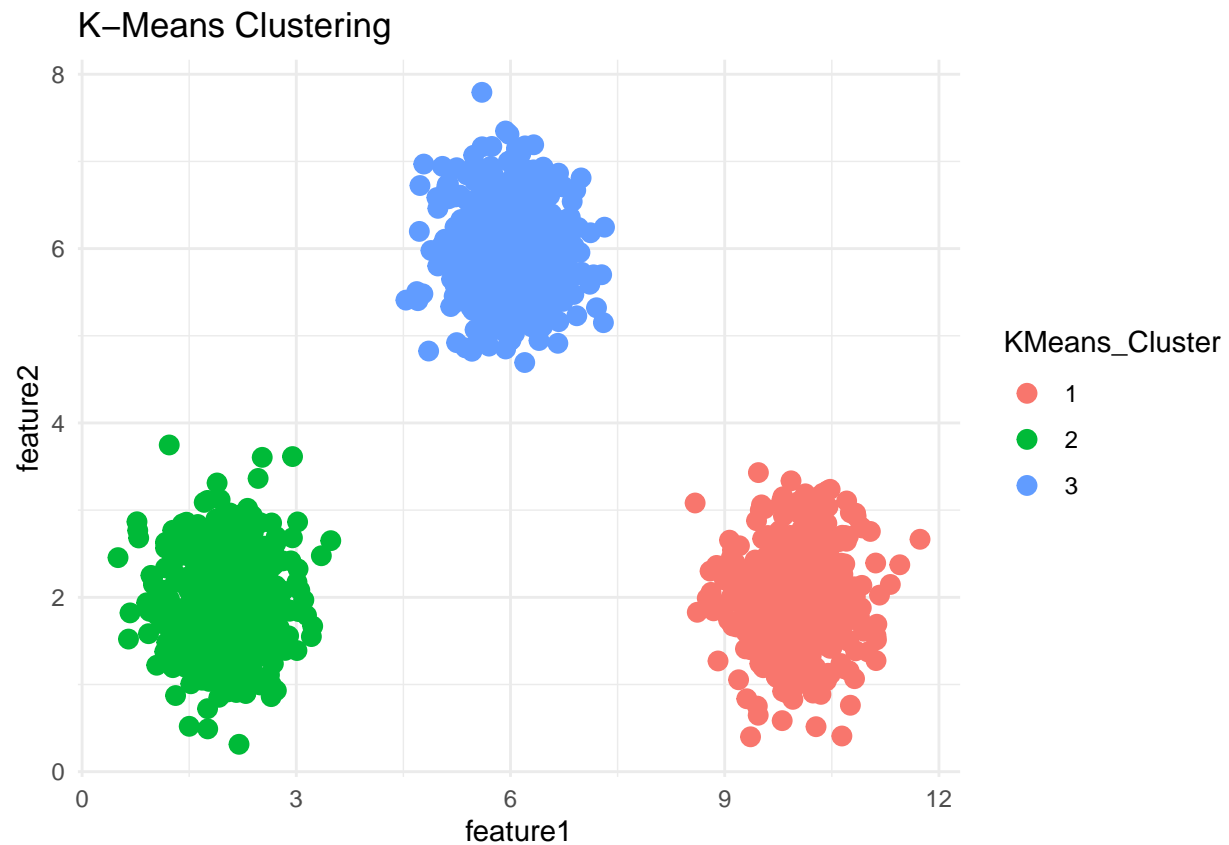```
K-Medoids: 0.08 sec elapsed
```

## Prep data for visualization

```
data$KMeans_Cluster <- as.factor(kmeans_result$cluster)
data$KMedoids_Cluster <- as.factor(kmedoids_result$clustering)
```

## Plotting k-means clusters

```
p1 <- ggplot(data, aes(x = feature1, y = feature2, color = KMeans_Cluster)) +
  geom_point(size = 3) +
  ggtitle("K-Means Clustering") +
  theme_minimal()

# Print the K-Means plot
print(p1)
```

K–Means Clustering

Plotting k-medoids clusters

```r
p2 <- ggplot(data, aes(x = feature1, y = feature2, color = KMedoids_Cluster)) +
  geom_point(size = 3) +
  ggtitle("K-Medoids Clustering") +
  theme_minimal()

# Print the K-Medoids plot
print(p2)
```

K–Medoids Clustering
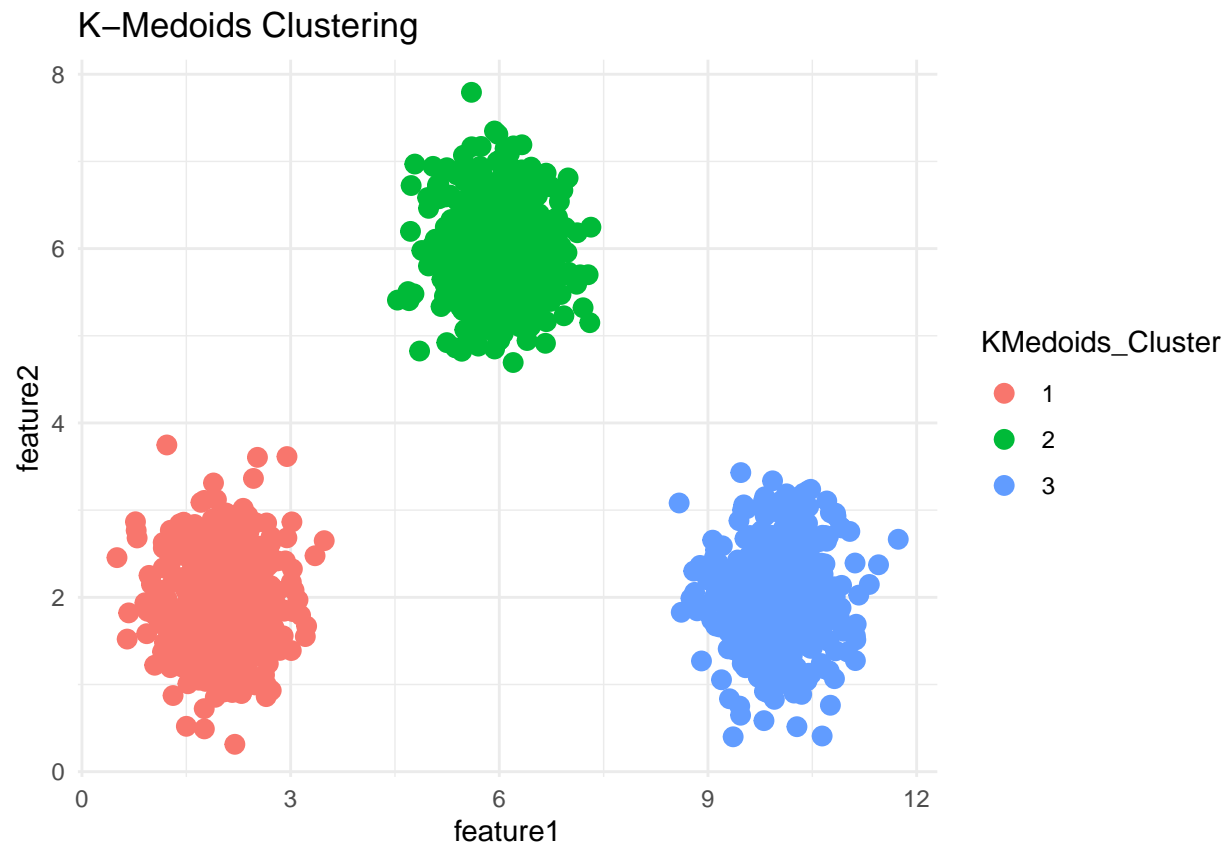
Load and prepare iris dataset

```
data(iris)

# Check available features
# colnames(iris)

df <- iris[, c("Petal.Length", "Petal.Width")]
```

Apply k-means and k-medoids

```
# K-means
tic("K-Means Iris")  # Start timer
kmeans_iris_result <- kmeans(df, centers = 3, nstart = 25)
toc()
```

```
## K-Means Iris: 0 sec elapsed
```

```r
# K-medoids
tic("K-medoids")
kmedoids_iris_result <- pam(df, k = 3)
toc()
```

```
## K-medoids: 0 sec elapsed
```

Prep for visualization

```r
df$KMeans_Cluster <- as.factor(kmeans_iris_result$cluster)
df$KMedoids_Cluster <- as.factor(kmedoids_iris_result$clustering)
df$Species <- iris$Species
```

Plot k-means

```r
p3 <- ggplot(df, aes(x = Petal.Length, y = Petal.Width, color = KMeans_Cluster, shape = Species)) +
  geom_point(size = 3) +
  ggtitle("Iris K-Means Clustering") +
  theme_minimal()

# Print the K-Means plot
print(p3)
```

Iris K−Means Clustering

Plot k-medoids

```
p4 <- ggplot(df, aes(x = Petal.Length, y = Petal.Width, color = KMedoids_Cluster, shape = Species)) +
  geom_point(size = 3) +
  ggtitle("Irirs K-Medoids Clustering") +
  theme_minimal()

# Print the K-Medoids plot
print(p4)
```

Irirs K–Medoids Clustering