Tutorial K-Means

Tim Fundamental Sains Data

11/17/2020

Tutorial ini adalah bagian dari mata kuliah Fundamental Sains Data, Informatika, UII, disusun sebagian berdasarkan contoh yang ada di sini.

Contoh A

Pada contoh ini, kita akan menggunakan data iris yang ada di package datasets. Untuk itu, jika belum meng-install package tersebut, silakan install terlebih dahulu.

```
library("datasets")
data("iris")
summary(iris)
```

```
##   Sepal.Length    Sepal.Width     Petal.Length    Petal.Width
##  Min.   :4.300   Min.   :2.000   Min.   :1.000   Min.   :0.100
##  1st Qu.:5.100   1st Qu.:2.800   1st Qu.:1.600   1st Qu.:0.300
##  Median :5.800   Median :3.000   Median :4.350   Median :1.300
##  Mean   :5.843   Mean   :3.057   Mean   :3.758   Mean   :1.199
##  3rd Qu.:6.400   3rd Qu.:3.300   3rd Qu.:5.100   3rd Qu.:1.800
##  Max.   :7.900   Max.   :4.400   Max.   :6.900   Max.   :2.500
##        Species
##  setosa    :50
##  versicolor:50
##  virginica :50
##
##
##
```

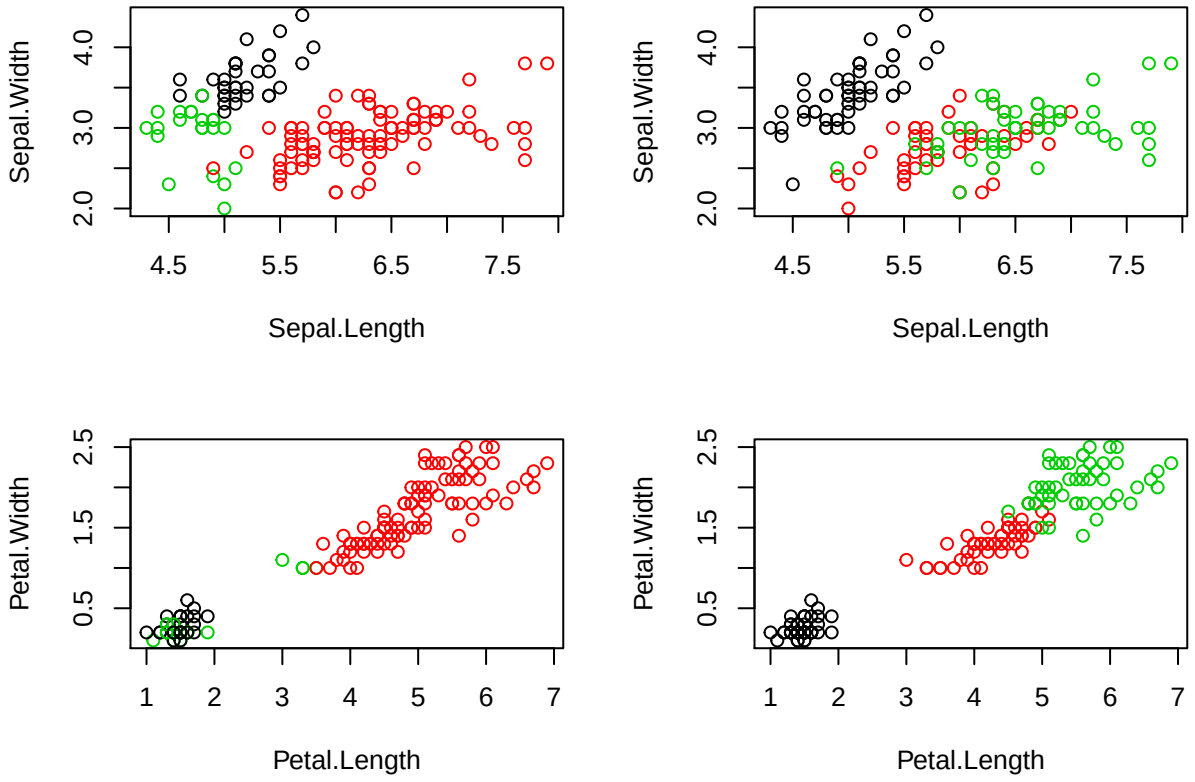
Pada contoh ini, kita akan melakukan clustering menggunakan K-Means pada data set bunga iris, berdasarkan empat variabel pertama, yaitu `Sepal.Length`, `Sepal.Width`, `Petal.length`, dan `Petal.Width`. Di sini, kita tidak akan menggunakan variabel kelas Species dan mencari cluster.

```
iris.new <- iris[, 1:4]
iris.class <- iris[, "Species"]
set.seed(1356)
result <- kmeans(iris.new, 3)
```

Secara spesifik, contoh di atas mencari 3 cluster dari data set `iris` (lihat baris ke-4). Selanjutnya, kita akan membuat beberapa plot untuk memvisualisasikan hasil clustering menggunakan K-Means.

Perhatikan script di bawah ini, baris ke-1, parameter `mfrow=c(2,2)` membagi ruang plot menjadi 2 baris dan 2 kolom, yang artinya ada 4 plot yang mungkin untuk ditampilkan bersama. Adapun parameter `mar=c(5,4,2,2)` mengatur ukuran margin dari plot.

```
par(mfrow=c(2,2), mar=c(5,4,2,2))
plot(iris.new[, c(1,2)], col=result$cluster)
plot(iris.new[, c(1,2)], col=iris.class)
plot(iris.new[, c(3,4)], col=result$cluster)
plot(iris.new[, c(3,4)], col=iris.class)
```



Plot pada kolom di sebelah kiri adalah hasil clustering menggunakan K-Means, sedangkan plot pada kolom sebelah kanan adalah plot spesies bunga iris berdasarkan dua variabel; `Sepal.Width-Sepal.Length` (plot di kanan-atas), `Petal.Width-Petal.Length` (plot di kanan bawah).

Perhatikan dua plot pada baris pertama, dapat kita lihat bahwa hasil clustering (kiri-atas) merepresentasikan kelompok spesies dengan cukup akurat jika kita bandingkan dengan plot spesies pada data yang asli (kanan-atas). Tiga spesies direpresentasikan dengan tiga warna yang berbeda: hitam, hijau dan merah. Adapun urutan warna yang tidak sama pada hasil clustering dan plot spesies pada data aslinya, tidak menjadi masalah, umpama, kelompok spesies yang direpresentasikan warna hijau di kiri-atas diplot dengan warna merah di kanan-atas.

Plot di baris kedua juga menunjukkan hasil clustering yang cukup baik.

Di materi yang kita bahas di kelas, kita mengevaluasi **cost** function yang disebut sebagai **distortion** function. Di sini nilai distortion pada iterasi terakhir dapat dilihat melalui `tot.withinss`.

```
result$tot.withinss
```

```
## [1] 142.7535
```

Untuk melihat nilai apa saja yang dikembalikan oleh fungsi `kmeans`, dapat dilihat melalui

```
attributes(result)$names
```

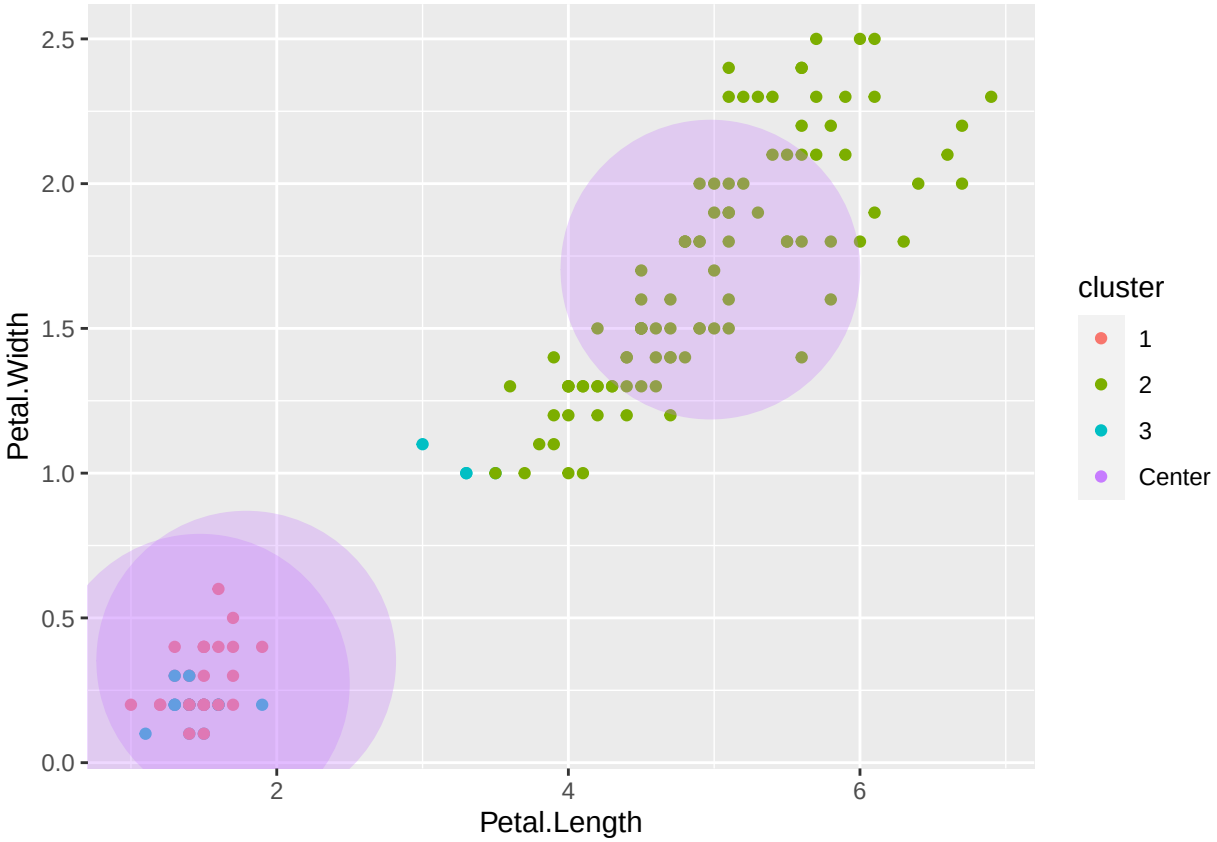
```
## [1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss"
## [6] "betweenss"    "size"         "iter"         "ifault"
```

dan untuk memanggil nilai tertentu di atas, dapat dilakukan dengan cara `result$cluster`, `result$size`, dan seterusnya. Untuk melihat deskripsi dari setiap nilai di atas dapat dilakukan dengan memanggil script `?kmeans` pada console.

Baris ke-4 hingga ke-6 adalah plotting menggunakan `ggplot2`. Di sini kita mencoba memberikan efek *radius* atau jangkauan dari setiap cluster.

Visualisasi menggunakan `ggplot2` Di bawah ini, kita akan memvisualisasikan hasil clustering di atas, menggunakan package `ggplot2`. Baris ke-1 menjadikan tipe data `cluster` dari numerik ke diskret (kategori/factor). Bagian ini hanyalah untuk kepentingan teknis.

```
iris.new$cluster <- factor(result$cluster)
centers <- as.data.frame(result$centers)
library(ggplot2)
ggplot() +
  geom_point(data=iris.new, aes(x=Petal.Length, y=Petal.Width,color=cluster))  +
  geom_point(data=centers, aes(x=Petal.Length, y=Petal.Width, color="Center"),
             size=52, alpha=.3, show.legend = FALSE)
```

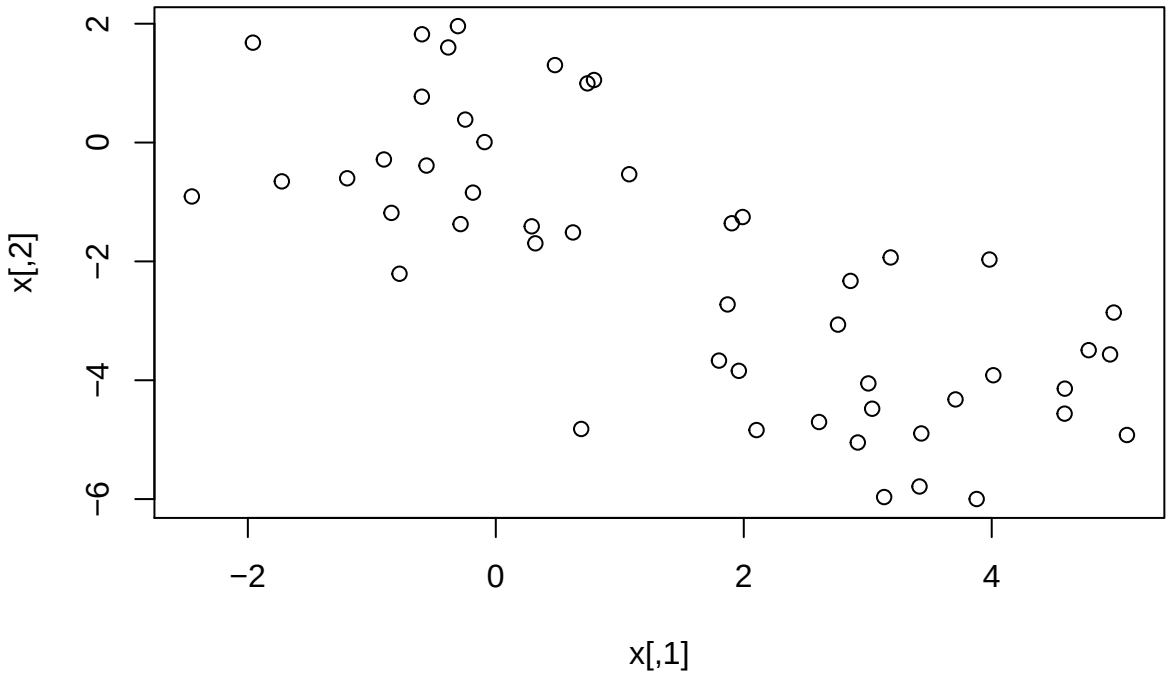


Untuk contoh lain visualisasi hasil clustering menggunakan ggplot, silakan lihat di sini

Contoh B

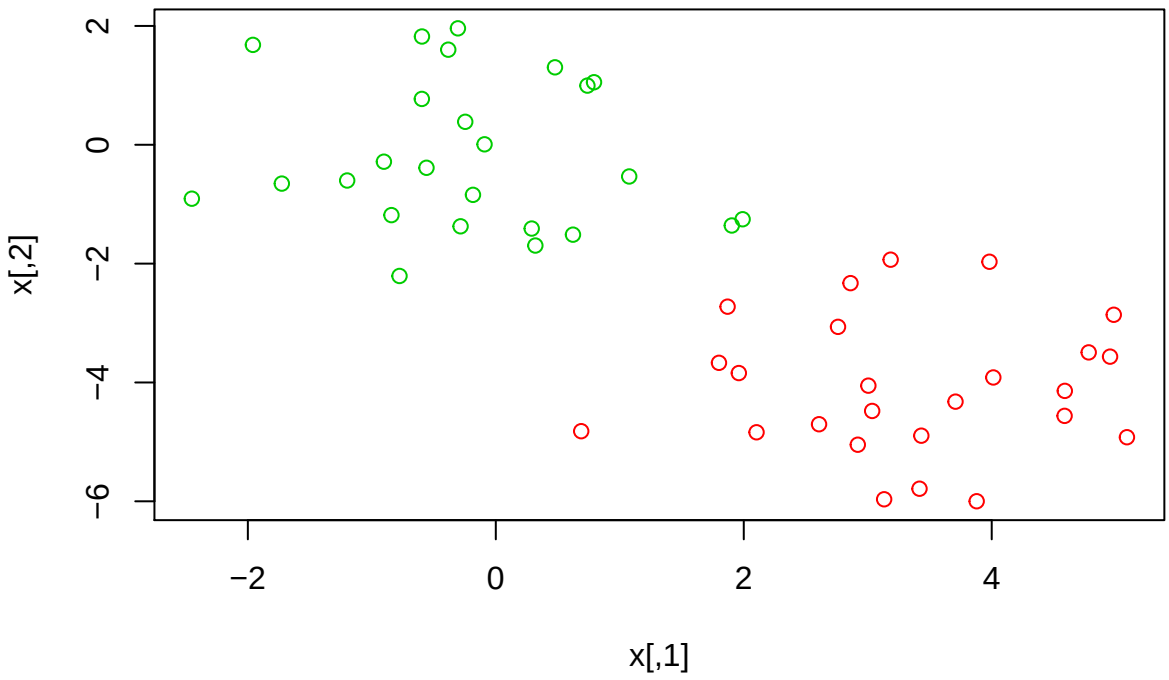
Pada contoh ini, kita akan mensimulasikan sebuah data set yang direpresentasikan dengan sebuah matriks yang terdiri dari 50 baris dan 2 kolom (Baris ke-2). Elemen dari matriks ini disimulasikan dari sebuah distribusi normal melalui fungsi `rnorm`. Baris ke-3 menambahkan setiap nilai pada kolom ke-1 dengan 3 dan baris ke-4 ,mengurangi setiap nilai pada kolom ke-2 dengan 4. Hal ini dilakukan untuk mensimulasikan sebuah sebaran data yang terdiri dari dua kelompok.

```
set.seed (2)
x <- matrix (rnorm (50*2) , ncol =2)
x[1:25 ,1] <- x[1:25 ,1] + 3
x[1:25 ,2]<- x[1:25 ,2] - 4
plot(x)
```



Dari plot di atas, dapat kita lihat, secara umum ada dua cluster. Selanjutnya, menggunakan K-Means, kita mencari dua cluster.

```
km.out <- kmeans(x, 2)
plot(x, col = (km.out$cluster +1))
```



Baris ke-1 menggunakan fungsi `kmeans` untuk mencari 2 cluster. Baris ke-2 mevisualisasikan hasil clustering dengan dua warna yang berbeda yang mengindikasikan dua cluster. Hasil plot dapat dilihat pada gambar di atas.