TUGAS BAGIAN II

SK-5222 PENAMBANGAN DATA DALAM SAINS

Mohammad Rizka Fadhli - 20921004

21 Mei 2022

Contents

List of Figures

List of Tables

SOAL DAN PEMBAHASAN

Soal I

Diberikan 10 buah titik data sebagai berikut:

Table 1: Data Soal I

titik	x	y
p1	4.0	5.2
p2	2.1	3.9
p3	3.4	3.1
p4	2.7	2.0
p5	0.8	4.1
p6	4.6	2.9
p7	4.3	1.2
p8	2.2	1.0
p9	4.1	4.1
p10	1.5	3.0

- Lakukan klasterisasi dari data tersebut dengan menggunakan algoritma *k-means* dengan jumlah partisi $K = 2$ sebanyak 10 kali.
- Tentukan sentroid awal (secara *random*) yang berbeda setiap melakukan klasterisasi.
- *Stopping criteria* untuk klasterisasi bisa ditentukan sendiri (tidak harus sampai tidak ada perubahan sentroid)

Pertanyaan

1. Tuliskan hasil akhir kluster yang didapat untuk setiap klasterisasi!
2. Hitung nilai *average* ***SSE*** untuk masing-masing hasil klusterisasi!
3. Hitung nilai *average* ***Sillhouette Coefficient*** untuk masing-masing hasil klusterisasi!
4. Dari hasil ***SSE*** dan ***Sillhouette Coefficient***, menurut Anda, hasil klasterisasi mana yang memberikan hasil terbaik? Berikan alasannya!
5. Apakah algoritma *K-means* sudah memberikan hasil yang baik? Apa yang dapat dilakukan agar hasil klasterisasi lebih baik?

Pembahasan

Untuk melakukan *k-means clustering* ini, saya akan membuat algoritma sendiri dengan menggunakan 2 titik *random* dan akan dilakukan sebanyak 10 kali. Berikut ini adalah *flowchart* dari algoritma tersebut:

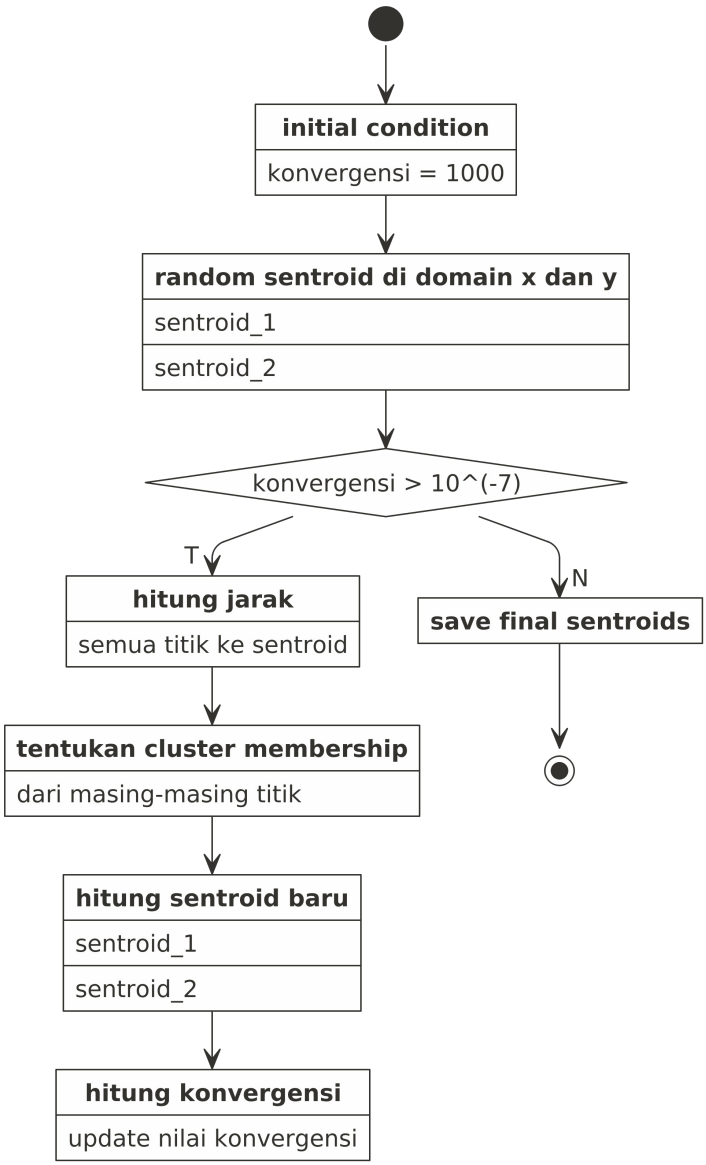

Figure 1: Flowchart Pengerjaan K-Means Clustering

Sebagai pengingat, algoritma *k-means clustering* dilakukan secara iteratif dengan mengandalkan suatu *stopping criteria* tertentu. Pada tugas ini, *stopping criteria* yang saya gunakan adalah sebagai berikut:

$$\text{konvergensi} = \sqrt{(x_1^{(k+1)} - x_1^{(k)})^2 + (x_2^{(k+1)} - x_2^{(k)})^2}$$

Dimana $x_1^{(k)}$ dan $x_2^{(k)}$ menandakan sentroid 1 dan 2 pada iterasi ke - k.

Sebagai gambaran, berikut adalah *scatterplot* dari data soal tersebut:

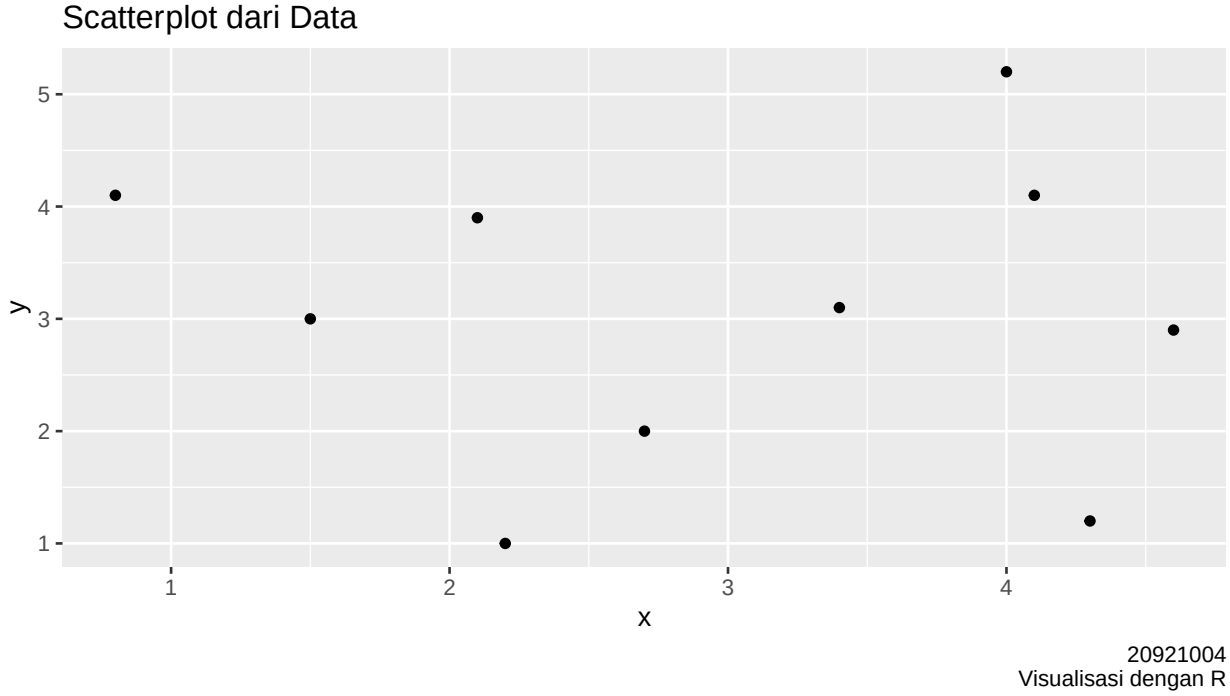

Figure 2: Scatterplot dari Data

Untuk menyelesaikan permasalahan ini, saya akan membuat beberapa program sebagai berikut:

Program untuk membuat sepasang titik secara *random*

```
# program untuk membuat titik sentroid secara random
random_titik = function(){
  list(
    sentroid_1 = runif(2,0,6),
    sentroid_2 = runif(2,0,6)
  )
}
```

Program menghitung *euclidean distance*

```
# program untuk menghitung jarak
jarak = function(x1,x2){
  sb_1 = (x1[1] - x2[1])^2
  sb_2 = (x1[2] - x2[2])^2
  sqrt(sb_1 + sb_2)
}
```

Program untuk menghitung sentroid baru hasil iterasi ke-i

```
# program untuk menghitung sentroid baru
new_sentroid = function(data){
  hit =
    data %>%
    group_by(cluster_no) %>%
    summarise(x = mean(x),
              y = mean(y)) %>%
    ungroup()
  output = list(sentroid_1 = c(hit$x[1],hit$y[1]),
                sentroid_2 = c(hit$x[2],hit$y[2]))
  return(output)
}
```

Program untuk menghitung konvergensi

```
# program untuk menghitung selisih sentroid baru dengan sentroid lama
konvergen_yn = function(){
  part1 = sentroid_baru$sentroid_1 - sentroid_1
  part2 = sentroid_baru$sentroid_2 - sentroid_2
  sqrt(sum(part1^2) + sum(part2^2))
}
```

Program untuk menghitung SSE

```
hitung_SSE = function(df){
  # hitung jarak terhadap sentroid
  for(i in 1:nrow(df)){
    titik = c(df$x[i],df$y[i])
    df$jarak_sentroid1[i] = jarak(titik,sentroid_1)
    df$jarak_sentroid2[i] = jarak(titik,sentroid_2)
  }
  SSE_final =
    df %>%
    mutate(jarak_thd_centroid = ifelse(cluster_no == 1,
                                       jarak_sentroid1,
                                       jarak_sentroid2))
  SSE_final$jarak_thd_centroid^2 %>% sum() %>% round(4)
}
```

***K-Means Clustering* pada pasangan titik *random* I**

Menggunakan titik random berikut ini:

```
random = random_titik()
random
```

```
## $sentroid_1
## [1] 2.303229 5.946158
##
## $sentroid_2
## [1] 5.666725 1.667516
```

Saya akan lakukan *clustering* berikut ini:

```r
# initial sentroid
sentroid_1 = random$sentroid_1
sentroid_2 = random$sentroid_2
# menyiapkan template untuk menghitung jarak
df$jarak_sentroid1 = NA
df$jarak_sentroid2 = NA
# initial konvergensi
konvergensi = 1000
# proses iterasi k-means hingga konvergensi tercapai
while(konvergensi > 10^(-7)){
  # hitung jarak terhadap sentroid
  for(i in 1:nrow(df)){
    titik = c(df$x[i],df$y[i])
    df$jarak_sentroid1[i] = jarak(titik,sentroid_1)
    df$jarak_sentroid2[i] = jarak(titik,sentroid_2)
  }
  # memasukkan masing-masing titik ke cluster terdekat
  df =
    df %>%
    mutate(cluster_no = ifelse(jarak_sentroid1 < jarak_sentroid2,1,2))
  # menghitung sentroid baru
  sentroid_baru = new_sentroid(df)
  # menghitung konvergensi
  konvergensi = konvergen_yn()
  # update sentroid baru
  sentroid_1 = sentroid_baru$sentroid_1
  sentroid_2 = sentroid_baru$sentroid_2
}
```

```r
# hasil final
df_1
```

```
##    titik   x   y jarak_sentroid1 jarak_sentroid2 cluster_no
## 1     p1 4.0 5.2       1.8840382       3.2092367          1
## 2     p2 2.1 3.9       0.4308132       2.2924223          1
## 3     p3 3.4 3.1       1.3159027       1.0607544          2
## 4     p4 2.7 2.0       2.0696860       0.7410803          2
## 5     p5 0.8 4.1       1.7004705       3.3486117          1
## 6     p6 4.6 2.9       2.3990832       1.4440222          2
## 7     p7 4.3 1.2       3.3792899       1.2021647          2
## 8     p8 2.2 1.0       3.0746707       1.6183943          2
## 9     p9 4.1 4.1       1.6004999       2.1631459          1
## 10   p10 1.5 3.0       1.4572577       2.1645323          1
```

```r
# sentroid 1
sentroid_1
```

```
## [1] 2.50 4.06
```

```r
# sentroid 2
sentroid_2
```

```
## [1] 3.44 2.04
```

Berikut adalah ***SSE*** dari perhitungan ini:

```r
# menghitung SSE
SSE = hitung_SSE(df_1)
SSE
```

```
## [1] 19.136
```

Soal II

Diberikan *confusion matrix* sebagai berikut:

Table 2: Data Soal II

cluster	entertainment	financial	foreign	metro	national	sports	Total
#1	1	1	0	11	4	676	693
#2	27	89	333	827	253	33	1562
#3	326	465	8	105	16	29	949
Total	354	555	341	943	273	738	3204

Pertanyaan

Hitung nilai *entropy* dan *purity* untuk matriks tersebut! Berikan analisis untuk hasil yang didapat!

Pembahasan

Entropi untuk masing-masing cluster dihitung sebagai berikut:

$$\begin{aligned}\text{Entropy 1} = & -\tfrac{1}{693}\log_2(\tfrac{1}{693}) - \tfrac{1}{693}\log_2(\tfrac{1}{693}) \\ & -0 - \tfrac{11}{693}\log_2(\tfrac{11}{693}) \\ & -\tfrac{4}{693}\log_2(\tfrac{4}{693}) - \tfrac{676}{693}\log_2(\tfrac{676}{693}) \\ & = 0.200\end{aligned}$$

$$\begin{aligned}\text{Entropy 2} = & -\tfrac{27}{1562}\log_2(\tfrac{27}{1562}) - \tfrac{89}{1562}\log_2(\tfrac{89}{1562}) \\ & -\tfrac{333}{1562}\log_2(\tfrac{333}{1562}) - \tfrac{872}{1562}\log_2(\tfrac{872}{1562}) \\ & -\tfrac{253}{1562}\log_2(\tfrac{253}{1562}) - \tfrac{33}{1562}\log_2(\tfrac{33}{1562}) \\ & = 1.841\end{aligned}$$

$$\begin{aligned}\text{Entropy 3} = & -\tfrac{326}{949}\log_2(\tfrac{326}{949}) - \tfrac{465}{949}\log_2(\tfrac{465}{949}) \\ & -\tfrac{8}{949}\log_2(\tfrac{8}{949}) - \tfrac{105}{949}\log_2(\tfrac{105}{949}) \\ & -\tfrac{16}{949}\log_2(\tfrac{16}{949}) - \tfrac{29}{949}\log_2(\tfrac{29}{949}) \\ & = 1.696\end{aligned}$$

Sedangkan untuk *purity* dihitung dengan cara:

$$\begin{aligned}\text{Purity 1} &= \tfrac{676}{693} = 0.975 \\ \text{Purity 2} &= \tfrac{827}{1562} = 0.529 \\ \text{Purity 3} &= \tfrac{465}{949} = 0.490\end{aligned}$$

Total entropy dihitung sebagai berikut:

$$\text{Total entropy} = \frac{693 \times 0.200 + 1562 \times 1.841 + 949 \times 0.490}{3204} = 0.614$$

Total purity dihitung sebagai berikut:

$$\text{Total purity} = \frac{693 \times 0.975 + 1562 \times 0.529 + 949 \times 1.696}{3204} = 1.443$$

Berikut jika disajikan dalam bentuk tabel:

Table 3: Hasil Perhitungan Entropy dan Purity

cluster	entertainment	financial	foreign	metro	national	sports	Total	Entropy	Purity
#1	1	1	0	11	4	676	693	0.200	0.975
#2	27	89	333	827	253	33	1562	1.841	0.529
#3	326	465	8	105	16	29	949	1.696	0.490
Total	354	555	341	943	273	738	3204	0.614	1.443

Dari tabel di atas, kita bisa dapatkan informasi sebagai berikut:

> *Cluster* `#1` memiliki *purity* yang sangat tinggi dan *entropy* terendah. Artinya, cluster ini berhasil mengelompokkan data yang *unique* karakteristiknya (berasal dari satu atribut dominan). Berbeda dengan *cluster* `#2` dan `#3` yang tidak memiliki satu atribut yang dominan. Tapi secara keseluruhan, *cluster* yang dihasilkan sudah bisa memisahkan data menjadi 3 kelompok dengan karakteristik yang berbeda-beda.