MO433 – Aprendizado de Máquina Não Supervisionado

Trabalho 2

Thiago Bruschi Martins RA 120212

Obs: Este relatório foi desenvolvido utilizando a linguagem R no RStudio.

```
library(factoextra)
```

```
## Carregando pacotes exigidos: ggplot2
```

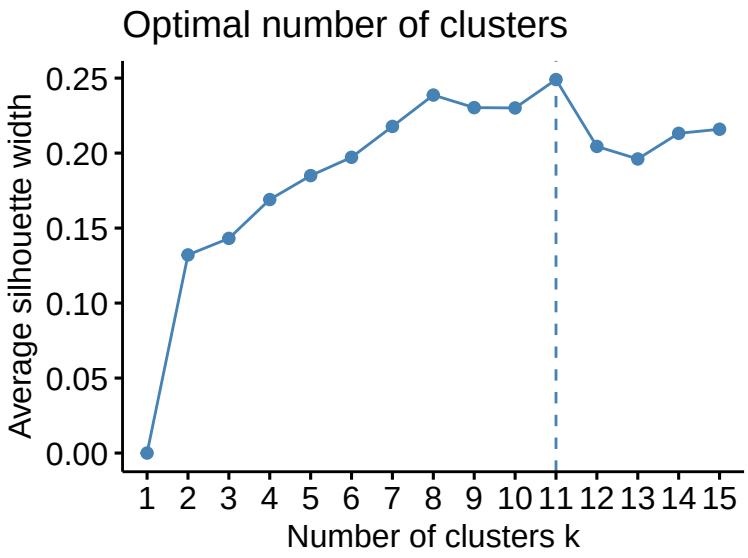
```
## Welcome! Want to learn more? See two factoextra-related books at https://goo.gl/ve3WBa
```

```
options(warn = -1)

# Setando o workspace
setwd("C:\\Users\\thiag\\OneDrive\\Documents\\Unicamp\\Master\\Unsupervised-Learning\\Trabalho 2")

# Leitura dos dados
data <- read.csv('ex2-data.csv', sep = "", header = FALSE)
```

1) K-means

Utilizando a função *fviz_nbclust* da biblioteca *factoextra* podemos escolher o algoritmo, o range do K e a métrica de avaliação dos modelos. Obtemos ainda um gráfico que ilustra o desempenho de cada modelo para cada valor de k. A seguir temos o gráfico gerado utilizando a silhueta como métrica.

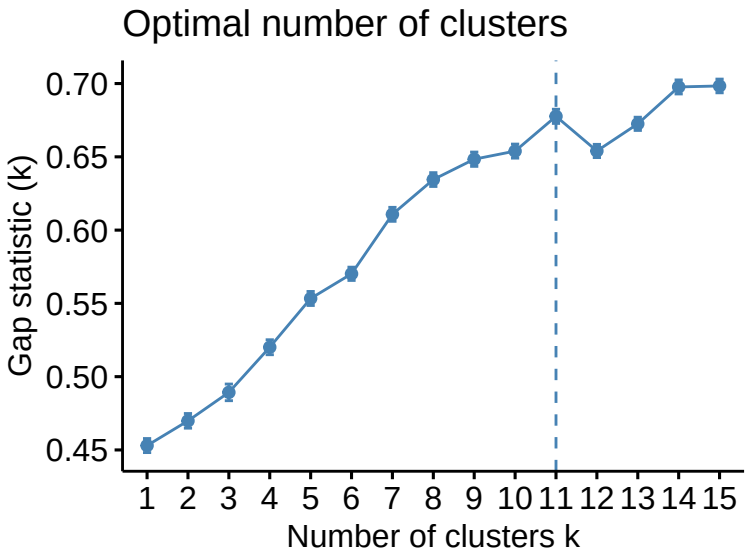
```
fviz_nbclust(data, kmeans, method = "silhouette", k.max=15)
```



A silhueta é uma métrica que usa uma noção de distância intra-cluster e inter-cluster. Como queremos uma distância intra-cluster pequena e uma distância inter-cluster grande, isso significa que queremos uma silhueta maior possível. Observando o gráfico acima, gerado pelos nossos modelos, vemos que o maior valor de silhueta é alcançado quando K = 11. Desta forma, para esta métrica e para este range, o melhor valor de K é 11, como indicado na figura.

Vamos agora analisar o mesmo conjunto de dados e modelos, mas utilizando a métrica Gap statistics.

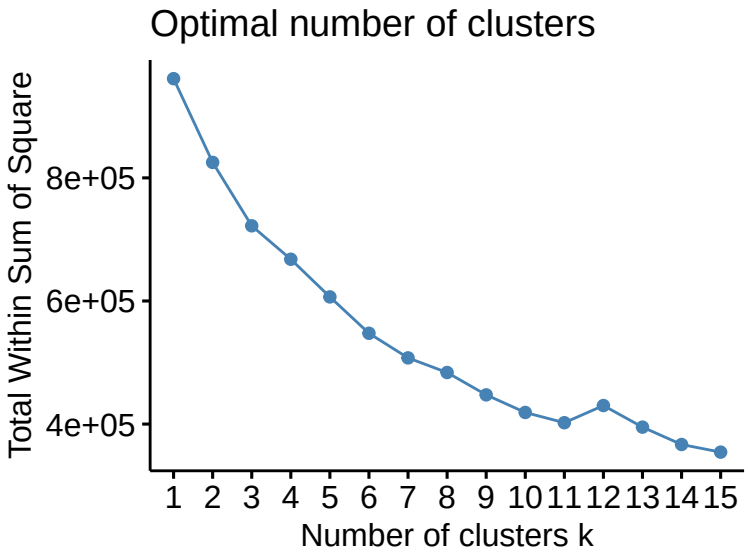
```
fviz_nbclust(data, kmeans, method = "gap_stat", k.max=15)
```



Esta métrica não foi vista em aula, porém tem-se que o melhor valor de K é quando o valor do *Gap*(K+1) está a uma distância menor que 1 std do *Gap*(K). E é exatamente isso que indica a função *fviz_nbclust*. Pois o *Gap*(K=12) não só deixa de aumentar como chega a ser menor que o *Gap*(K=11). Sendo assim, de acordo com esta métrica, o melhor valor de K, dentro deste range, também é 11.

Por último, vamos avaliar nossos clusters utilizando a métrica WSS.

```
fviz_nbclust(data, kmeans, method = "wss", k.max=15)
```



A *WSS* nada mais é que a soma quadrática das distâncias entre cada ponto e o centróide de seu respectivo cluster. Para esta métrica deveríamos escolher o K onde há uma variação brusca no decrescimento do *WSS*, o chamado "cotovelo" da curva. No entanto, na curva gerada nós não encontramos exatamente um cotovelo. Mas podemos ver que para K=12 o *WSS* não só para de descer, como dá uma subida inesperada. Assim, considerando que as outras duas métricas apontaram para K=11 como o melhor valor, podemos considerar que K=11 também é um bom valor para esta métrica, pois há uma mudança muito brusca no comportamento (ou derivada) do *WSS* para K=12.

2) Escolha do K

Como analisado em cada gráfico, os resultados das 3 métricas convergiram. Mesmo considerando que o resultado da WSS não é muito claro, podemos dizer que K=11 é um ótimo valor para esta métrica também. Sendo assim, o número de clusters escolhido é 11.

3) GMM

Vamos agora realizar a clusterização utilizando modelos GMM e o melhor K encontrado nos modelos anteriores, criamos 3 modelos GMM com restrições distintas sobre o formato dos clusters, que são modelados como gaussianas.

```
library('mclust')
```

```
## Package 'mclust' version 5.4.8
## Type 'citation("mclust")' for citing this R package in publications.
```

```
# Criando os 3 modelos com restrições distintas
spherical <- Mclust(data, G=11, modelNames = 'VII')
diagonal <- Mclust(data, G=11, modelNames = 'VVI')
full <- Mclust(data, G=11, modelNames = 'VVV')
```

4) Comparando as clusterizações

Agora utilizaremos uma função que computa várias métricas para comparação das clusterizações, no entanto, vamos focar em apenas duas:

1) Rand Index Corrigido (adjusted)
2) Jaccard Index

```
library('mclustcomp')
# Computa as metricas de comparação entre os modelos
full_sphe <- mclustcomp(full$classification, spherical$classification)
full_diag <- mclustcomp(full$classification, diagonal$classification)

# Gerando o dataframe de saida
modelos_comparados <- c('GMM Full x GMM Spherical', 'GMM Full x GMM Diagonal',
                        'GMM Full x GMM Spherical', 'GMM Full x GMM Diagonal')
metrica <- c('Adjusted Rand Index', 'Adjusted Rand Index',
             'Jaccard Index', 'Jaccard Index')
score <- c(full_sphe[full_sphe$types=='adjrand',2],
           full_diag[full_diag$types=='adjrand',2],
           full_sphe[full_sphe$types=='jaccard',2],
           full_diag[full_diag$types=='jaccard',2])
df <- data.frame(modelos_comparados, metrica, score)
print(df)
```

```
##         modelos_comparados             metrica     score
## 1 GMM Full x GMM Spherical Adjusted Rand Index 0.6652865
## 2  GMM Full x GMM Diagonal Adjusted Rand Index 0.8285284
## 3 GMM Full x GMM Spherical       Jaccard Index 0.5397065
## 4  GMM Full x GMM Diagonal       Jaccard Index 0.7323344
```

Considerando o GMM Full como "gabarito", podemos dizer que o modelo GMM Diagonal foi melhor do que o GMM Spherical, pois apresentou mais semelhança com o nosso "gabarito" nas duas métricas escollhidas. Esta seria uma maneira de utilizar métricas externas para avaliar diferentes clusters, isto é, comparando os resultados obtidos com dados externos.