# TP1 - Document clustering avec le modèle JOSE sur BBC-News dataset

Ce TP est réalisé dans le cadre d'évaluation du module mixture-models et co-clustering du master 2 MLDS - Machine Learning for Data Science.

Equipe :

- Abdesselam BENAMEUR
- Hakim IGUENI

# Install Packages

```
# install.packages("NbClust")
# # install.packages("factoextra")
# install.packages("skmeans")
# install.packages("movMF")
# install.packages("R.matlab")
# install.packages("aricode")
```

# Import Packages

```
library(NbClust)
# library(factoextra)
library(skmeans)
library(movMF)
library(R.matlab)
```

```
## R.matlab v3.7.0 (2022-08-25 21:52:34 UTC) successfully loaded. See ?R.matlab fo
r help.
```

```
##
## Attaching package: 'R.matlab'
```

```
## The following objects are masked from 'package:base':
##
##     getOption, isOpen
```

```
library(aricode)
```

# Load Data

```
# Le lien pour télécharger les deux formats de données ainsi que les labels.
url <- "https://cifre.s3.eu-north-1.amazonaws.com/bbc_dataset.mat"
# Définir le nom du fichier de destination.
# destination_file <- "bbc_dataset.mat"
# Télécharger le fichier.
# download.file(url, destfile = destination_file, method = "auto")
bbc_dataset <- readMat(url)
bbc_jose <- bbc_dataset$jose
bbc_doc_term <- bbc_dataset$doc.term
bbc_labels <- as.vector(bbc_dataset$labels)
```

```
dim(bbc_jose)
```

```
## [1] 2225  100
```

```
dim(bbc_doc_term)
```

```
## [1] 2225 2000
```

1. **Quelle est la dimension de chaque dataset (nombre documents, nombre de mots/taille de la dimension) ?**

- Le 1er dataset (la matrice document-terme, la variable bbc_doc_term)
    - nombre documents: 2225
    - nombre de mots: 2000
- Le 2eme dataset (la matrice des embeddings des documents, la variable bbc_jose)
    - nombre documents: 2225
    - taille de la dimension: 100

2. **Exécuter tous les algorithmes de clustering avec le vrai nombre de classes qui est 5.**
3. **Afficher les matrices de confusion obtenues à chaque application d'un algorithme de clustering.**
4. **Pour mesurer la qualité du clustering obtenu, on s'appuiera sur les mesures externes : NMI et ARI.**
5. **Commenter les résultats obtenus.**
6. ***Questions optionnelles :***

   ***a. Toute utilisation d'un algorithme de clustering supplémentaire sera apprécié.***

   ***b. Une analyse sur le nombre de clusters peut être envisagée à partir de vos connaissances.***

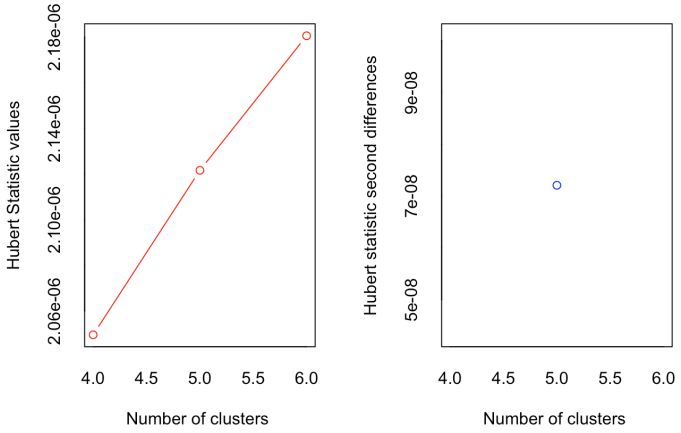
# Clustering of the first Dataset (bbc_doc_term)

# K-means

```
# we use nbclust, and k = 5
set.seed(123)
res.kmeans <- NbClust(bbc_doc_term, distance = "euclidean", min.nc = 4, max.nc =
6, method = "kmeans")
```

```
## Warning in max(DiffLev[, 3], na.rm = TRUE): no non-missing arguments to max;
## returning -Inf
```

```
## Warning in max(DiffLev[, 4], na.rm = TRUE): no non-missing arguments to max;
## returning -Inf
```

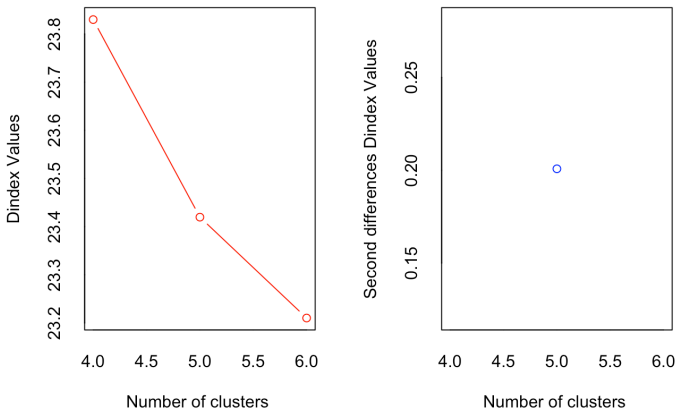



```
## *** : The Hubert index is a graphical method of determining the number of clust
ers.
##                 In the plot of Hubert index, we seek a significant knee that co
rresponds to a
##                 significant increase of the value of the measure i.e the signif
icant peak in Hubert
##                 index second differences plot.
##
```

```
## *** : The D index is a graphical method of determining the number of clusters.
##                 In the plot of D index, we seek a significant knee (the signifi
cant peak in Dindex
##                 second differences plot) that corresponds to a significant incr
ease of the value of
##                 the measure.
##
```

```
## Warning in matrix(c(results), nrow = 2, ncol = 26): data length [50] is not a
## sub-multiple or multiple of the number of columns [26]
```

```
## Warning in matrix(c(results), nrow = 2, ncol = 26, dimnames =
## list(c("Number_clusters", : data length [50] is not a sub-multiple or multiple
## of the number of columns [26]
```



```
## *****************************************************************
## * Among all indices:
## * 13 proposed 4 as the best number of clusters
## * 7 proposed 5 as the best number of clusters
## * 3 proposed 6 as the best number of clusters
##
##                     ***** Conclusion *****
##
## * According to the majority rule, the best number of clusters is  4
##
##
## *****************************************************************
```

```
# res.kmeans <- kmeans(bbc_doc_term, 5, nstart = 25)

# # we use fviz_cluster to visualize the clusters without points labels
# library(factoextra)
# fviz_cluster(res.kmeans, data = bbc_jose, ellipse.type = "norm",
#              palette = "jco", ggtheme = theme_minimal(),
#              main = "K-means clustering of documents")
```

```
# confusion matrix
clusters.kmeans <- res.kmeans$Best.partition
table(clusters.kmeans, bbc_labels)
```

```
##                bbc_labels
## clusters.kmeans   0   1   2   3   4
##               1 188 155 120  94 263
##               2   1   0   0   3   3
##               3  98  20  24  10  35
##               4 114 335 367 279 116
```

```
# NMI
nmi.kmeans <- NMI(clusters.kmeans, bbc_labels)
paste("NMI:", nmi.kmeans)
```

```
## [1] "NMI: 0.0628969449475587"
```

```
# ARI
ari.kmeans <- ARI(clusters.kmeans, bbc_labels)
paste("ARI:", ari.kmeans)
```

```
## [1] "ARI: 0.0633143450678484"
```

## Spherical K-means

```
# we use skmeans, and k = 5
set.seed(123)
res.skmeans <- skmeans(bbc_doc_term, 5)

# confusion matrix
clusters.skmeans <- res.skmeans$cluster
table(clusters.skmeans, bbc_labels)
```

```
##                bbc_labels
## clusters.skmeans   0   1   2   3   4
##                1 267 120  18  33 105
##                2  96 142 108 237  93
##                3  26 225  79  58  12
##                4  10  23   4  29 203
##                5   2   0 302  29   4
```

```
# NMI
nmi.skmeans <- NMI(clusters.skmeans, bbc_labels)
paste("NMI:", nmi.skmeans)
```

```
## [1] "NMI: 0.276233388100149"
```

```
# ARI
ari.skmeans <- ARI(clusters.skmeans, bbc_labels)
paste("ARI:", ari.skmeans)
```

```
## [1] "ARI: 0.212289829517698"
```

## von-Mises Fisher Mixture Model

```
# we use movMF, and k = 5
set.seed(123)
res.movMF <- movMF(bbc_doc_term, 5, kappa=list(common = TRUE), nruns=5, maxit=200)
clusters.movMF <- apply(res.movMF$P,1,which.max)

# confusion matrix
table(clusters.movMF, bbc_labels)
```

```
##              bbc_labels
## clusters.movMF   0   1   2   3   4
##              1  23 190  95  60  11
##              2 274 148  19  41 109
##              3   7  23   4  28 199
##              4   1   0 285  27   4
##              5  96 149 108 230  94
```

```
# NMI
nmi.movMF <- NMI(clusters.movMF, bbc_labels)
paste("NMI:", nmi.movMF)
```

```
## [1] "NMI: 0.261955530184438"
```

```
# ARI
ari.movMF <- ARI(clusters.movMF, bbc_labels)
paste("ARI:", ari.movMF)
```

```
## [1] "ARI: 0.191144431198362"
```

# Hierarchical Clustering

```
# we use hclust, and k = 5
res.hclust <- hclust(dist(bbc_doc_term), method = "ward.D2")

# confusion matrix
print("Confusion matrix")
```

```
## [1] "Confusion matrix"
```

```
clusters.hclust <- cutree(res.hclust, 5)
table(clusters.hclust, bbc_labels)
```

```
##                bbc_labels
## clusters.hclust   0   1   2   3   4
##               1 218 158 224 100 309
##               2  91 343 274 278  94
##               3  89   9  11   5  11
##               4   3   0   2   2   1
##               5   0   0   0   1   2
```

```
# NMI
nmi.hclust <- NMI(clusters.hclust, bbc_labels)
paste("NMI:", nmi.hclust)
```

```
## [1] "NMI: 0.074779266469138"
```

```
# ARI

ari.hclust <- ARI(clusters.hclust, bbc_labels)
paste("ARI:", ari.hclust)
```

```
## [1] "ARI: 0.0657793900465407"
```

# Summary

| Clustering algorithm | NMI | ARI |
|---|---|---|
| K-means | 0.0628969449475587 | 0.0633143450678484 |
| Spherical K-means | **0.276233388100149** | **0.212289829517698** |
| von-Mises Fisher Mixture Model | 0.261955530184438 | 0.191144431198362 |
| Hierarchical Clustering | 0.074779266469138 | 0.0657793900465407 |

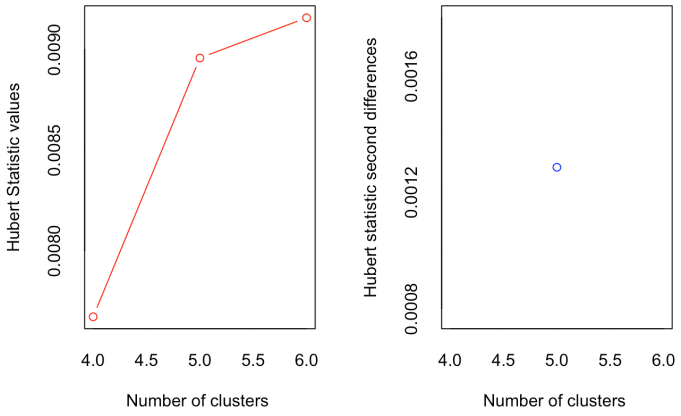
- The K-means algorithm shows relatively low values for both NMI and ARI. This suggests that the clusters generated by K-means do not align well with the true underlying structure of the data.
- Spherical K-means performs better than the traditional K-means, indicating that considering the spherical nature of the data space improves clustering quality. However, there is still room for improvement.
- The von-Mises Fisher Mixture Model demonstrates competitive performance, yielding higher NMI and ARI than the traditional K-means. It suggests that the model accounts for the distributional characteristics of the data better.
- Hierarchical clustering shows marginal improvement compared to K-means but still exhibits relatively low values for both NMI and ARI. This indicates that the hierarchical structure might not align well with the true clusters in the data.

In summary, none of the methods seem to perform exceptionally well, with NMI and ARI scores indicating suboptimal alignment with the true underlying structure of the BBC News dataset.

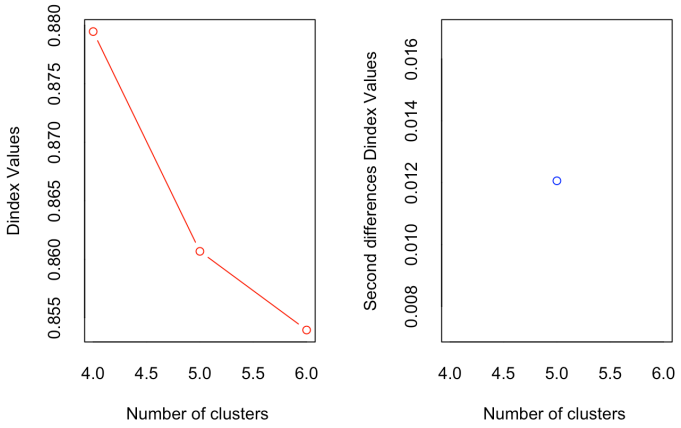
# Clustering of second Dataset (bbc_jose)

# K-means

```
# we use nbclust, and k = 5
set.seed(123)
res.kmeans <- NbClust(bbc_jose, distance = "euclidean", min.nc = 4, max.nc = 6, me
thod = "kmeans")
```




```
## *** : The Hubert index is a graphical method of determining the number of clust
ers.
##                 In the plot of Hubert index, we seek a significant knee that co
rresponds to a
##                 significant increase of the value of the measure i.e the signif
icant peak in Hubert
##                 index second differences plot.
##
```




```
## *** : The D index is a graphical method of determining the number of clusters.
##                 In the plot of D index, we seek a significant knee (the signifi
cant peak in Dindex
##                 second differences plot) that corresponds to a significant incr
ease of the value of
##                 the measure.
##
## *******************************************************************
## * Among all indices:
## * 4 proposed 4 as the best number of clusters
## * 15 proposed 5 as the best number of clusters
## * 4 proposed 6 as the best number of clusters
##
##                    ***** Conclusion *****
##
## * According to the majority rule, the best number of clusters is  5
##
##
## *******************************************************************
```

```
#res.kmeans <- kmeans(bbc_jose, 5, nstart = 25)

# # we use fviz_cluster to visualize the clusters without points labels
# library(factoextra)
# fviz_cluster(res.kmeans, data = bbc_jose, ellipse.type = "norm",
#              palette = "jco", ggtheme = theme_minimal(),
#              main = "K-means clustering of documents")
```

```
# confusion matrix
clusters.kmeans <- res.kmeans$Best.partition
table(clusters.kmeans, bbc_labels)
```

```
##                bbc_labels
## clusters.kmeans   0   1   2   3   4
##               1  10 481   3   1   9
##               2   1   0 501   1   4
##               3   9   3   5 370   2
##               4   4  15   2   4 396
##               5 377  11   0  10   6
```

```
# NMI
nmi.kmeans <- NMI(clusters.kmeans, bbc_labels)
paste("NMI:", nmi.kmeans)
```

```
## [1] "NMI: 0.857278886320236"
```

```
# ARI
ari.kmeans <- ARI(clusters.kmeans, bbc_labels)
paste("ARI:", ari.kmeans)
```

```
## [1] "ARI: 0.893826147337719"
```

En utilisant le package Nbclust avec plusieurs valeurs de k = {4, 5, 6} pour le nombre de clusters, on conclut que le meilleur clustering est celui avec K=5.

# Spherical K-means

```
# we use skmeans, and k = 5
set.seed(123)
res.skmeans <- skmeans(bbc_jose, 5, control = list(maxiter=200))

# confusion matrix
clusters.skmeans <- res.skmeans$cluster
table(clusters.skmeans, bbc_labels)
```

```
##                bbc_labels
## clusters.skmeans   0   1   2   3   4
##                1 375  11   0   9   6
##                2   5  16   2   4 396
##                3  11 479   3   6   9
##                4   9   3   4 366   2
##                5   1   1 502   1   4
```

```
# NMI
nmi.skmeans <- NMI(clusters.skmeans, bbc_labels)
paste("NMI:", nmi.skmeans)
```

```
## [1] "NMI: 0.84766649826647"
```

```
# ARI
ari.skmeans <- ARI(clusters.skmeans, bbc_labels)
paste("ARI:", ari.skmeans)
```

```
## [1] "ARI: 0.886328685712571"
```

# von-Mises Fisher Mixture Model

```
# we use movMF, and k = 5
set.seed(123)
res.movMF <- movMF(bbc_jose, 5, kappa=list(common = TRUE), nruns=5, maxit=200)
clusters.movMF <- apply(res.movMF$P,1,which.max)

# confusion matrix
table(clusters.movMF, bbc_labels)
```

```
##               bbc_labels
## clusters.movMF   0   1   2   3   4
##              1  12 479   3   6   9
##              2 373  11   0   9   6
##              3   5  16   2   4 396
##              4   2   1 502   1   4
##              5   9   3   4 366   2
```

```
# NMI
nmi.movMF <- NMI(clusters.movMF, bbc_labels)
paste("NMI:", nmi.movMF)
```

```
## [1] "NMI: 0.84511546069262"
```

```
# ARI
ari.movMF <- ARI(clusters.movMF, bbc_labels)
paste("ARI:", ari.movMF)
```

```
## [1] "ARI: 0.884235720048883"
```

# Hierarchical Clustering

```
# we use hclust, and k = 5
res.hclust <- hclust(dist(bbc_jose), method = "ward.D2")

# confusion matrix
print("Confusion matrix")
```

```
## [1] "Confusion matrix"
```

```
clusters.hclust <- cutree(res.hclust, 5)
table(clusters.hclust, bbc_labels)
```

```
##                bbc_labels
## clusters.hclust   0   1   2   3   4
##               1   1   1 508   1   3
##               2  14 460   1   2   8
##               3  13   6   0 373  10
##               4 372  11   0   3   3
##               5   1  32   2   7 393
```

```
# NMI
nmi.hclust <- NMI(clusters.hclust, bbc_labels)
paste("NMI:", nmi.hclust)
```

```
## [1] "NMI: 0.843993589277633"
```

```
# ARI

ari.hclust <- ARI(clusters.hclust, bbc_labels)
paste("ARI:", ari.hclust)
```

```
## [1] "ARI: 0.875001096159056"
```

# Summary

| Clustering algorithm | NMI | ARI |
|---|---|---|
| K-means | **0.857278886320236** | **0.893826147337719** |
| Spherical K-means | 0.84766649826647 | 0.886328685712571 |
| von-Mises Fisher Mixture Model | 0.84511546069262 | 0.884235720048883 |
| Hierarchical Clustering | 0.843993589277633 | 0.875001096159056 |

- K-means, when applied with JoSE embeddings, shows significantly higher NMI and ARI compared to the traditional doc-term matrix. This indicates that the Spherical text embeddings capture the inherent structure of the data more effectively, resulting in more coherent clusters.
- Spherical K-means continues to demonstrate strong performance with JoSE embeddings. The high NMI and ARI scores suggest that the Spherical embeddings contribute to better-defined clusters and improved alignment with the true structure of the data.
- Similar to K-means and Spherical K-means, the von-Mises Fisher Mixture Model exhibits robust clustering performance with JoSE embeddings. The model effectively captures the spherical distribution in the text embeddings, leading to accurate clustering.
- Hierarchical clustering with JoSE embeddings shows consistently strong results, with high NMI and ARI scores. This suggests that the hierarchical structure aligns well with the inherent clusters in the data, as captured by the Spherical text embeddings.

In summary, applying JoSE document embeddings significantly enhances the performance of clustering algorithms on the BBC News dataset. The higher NMI and ARI scores across all methods indicate that the embeddings better represent the semantic relationships within the text, leading to more accurate and cohesive clustering results.