Data in Public and Social Services - 3rd practical exercise class

15th April 2024
by Davide Chicco
davide.chicco@unimib.it

Practical exercise class lecturer: Vasco Coelho v.coelho@campus.unimib.it

The goals of this practical exercise class are the following:

A. Apply the concepts on clustering seen during the last class lecture on an EHRs dataset

R setup main commands

0) Put the following commands as header of your script:

```
setwd(".") #we use the current folder of the script as working directory
options(stringsAsFactors = FALSE) # we set the input strings not to be considered
set.seed(10) # we set a seed to be able to replicate our tests
options(repos = list(CRAN="http://cran.rstudio.com/")) # we set the URL
where to download the packages
```

1) load the pacman library for an easier installation and loading of the libraries:

```
# install.packages("pacman", dependencies = TRUE)
library("pacman")
```

# Load and/or install other packages we need

```
p_load("dlookr", "dplyr", "ggplot2",  "pastecs", "tableone", "umap",
"textshape", "factoextra", "ggdendro", "fpc", "cluster", "ggdendro",
"clusterSim", "parameters")
```

Application of the clustering methods k-means and hierarchical clustering to a dataset of electronic health records of patients with diabetes type 1 from Japan

2.1) Takashi 2019 diabetes type 1 dataset: we download the preprocessed version, that is the output of the first practical exercise class

Takashi2019_diabetes_type1_dataset_preprocessed.csv file to download

Takashi Y, Ishizu M, Mori H, Miyashita K, Sakamoto F, Katakami N, et al. (2019) “Circulating osteocalcin as a bone-derived hormone is inversely correlated with body fat in patients with type 1 diabetes”. PLOS ONE 14(5): e0216416. https://doi.org/10.1371/journal.pone.0216416

Cerono G, Chicco D. 2024, “Ensemble machine learning reveals key features for diabetes duration from electronic health records”. PeerJ Computer Science 10:e1896 https://doi.org/10.7717/peerj-cs.1896

A. Load the dataset

```
fileName <- "Takashi2019_diabetes_type1_dataset_preprocessed.csv"
patients_data <- read.csv(fileName, header = TRUE, sep =",")
```

B. Quantitative description

```
# We want to generate the descriptive statistics of all the features involved

patients_data %>% dim()
patients_data %>% summary()
patients_data %>% str()
patients_data %>% colnames() %>% sort()

patients_data %>% pastecs::stat.desc()

tableone::CreateTableOne(data=patients_data)

summary(tableone::CreateTableOne(data = patients_data))
```

C. Let’s prepare the dataset for k-means and then apply this method

```
# We need to scale the data so each feature has a mean of 0 and a standard deviation of 1

patients_data_scaled <- patients_data %>% scale() %>% as.data.frame()
patients_data_scaled %>% pastecs::stat.desc()

# k-means application

this_nstart <- 25 # number of initial random initializations
k_clusters <- 2
```

```
kmeans_results <- kmeans(patients_data_scaled, centers = k_clusters,
nstart = this_nstart)

kmeans_results %>% print()

# Understand the results of k-means application of the previous step

fviz_cluster(kmeans_results, data = patients_data_scaled)

# Save the output of the previous command into a PDF file, where you specified the
disease of the dataset, the clustering method employed, and the number of clusters in
the title

# we compute five evaluation statistics of this clustering application: Silhouette score,
Calinski-Harabasz index, Dunn index, Davies-Bouldin index, Gap statistic

# cluster.stats computes Silhouette score, Calinski-Harabasz index, and Dunn index

clustering_results_metrics <-
cluster.stats(d=dist(patients_data_scaled),
clustering=kmeans_results$cluster, silhouette=TRUE)

clustering_results_metrics %>% names() %>% sort()

cat("The average Silhouette score across the ", k_clusters, " clusters
generated by k-means is ", clustering_results_metrics$avg.silwidth, "
(the higher, the better)\n", sep="")

cat("The Calinski-Harabasz index with ", k_clusters, " clusters
generated by k-means is ", clustering_results_metrics$ch, " (the
higher, the better)\n", sep="")

cat("The Dunn index with ", k_clusters, " clusters generated by k-means
is ", clustering_results_metrics$dunn, " (the higher, the better)\n",
sep="")

# we compute the Davies-Bouldin index here
davies_bouldin_index_results <- index.DB(x=patients_data_scaled,
cl=kmeans_results$cluster)
davies_bouldin_index <- davies_bouldin_index_results$"DB"

cat("The Davies-Bouldin index with ", k_clusters, " clusters generated
by k-means is ",  davies_bouldin_index, " (the lower, the better)\n",
sep="")


# we compute the Gap statistic here
```

```
number_of_bootstrap_samples <- 60

gap_result <- clusGap(patients_data_scaled, FUN = kmeans, nstart =
this_nstart, K.max = k_clusters, B = number_of_bootstrap_samples,
verbose = FALSE)

average_gap <- as.data.frame(gap_result$"Tab")$"gap" %>% mean()

cat("The average Gap statistic across the ", k_clusters, " clusters
generated by k-means is ", average_gap, " (the higher, the better)\n",
sep="")
```

D. Repeat all these steps with 3 clusters and then determine which is the better number of clusters between 2 or 3 for k-means

E. Let’s prepare the dataset for hierarchical clustering and then apply this method

```
# We need to select the best linkage method

average_linkage_score <- agnes(patients_data_scaled,
method="average")$ac
single_linkage_score <- agnes(patients_data_scaled, method="single")$ac
complete_linkage_score <- agnes(patients_data_scaled,
method="complete")$ac
ward_linkage_score <- agnes(patients_data_scaled, method="ward")$ac

cat("average_linkage_score = ", average_linkage_score, "\n", sep="")
cat("single_linkage_score = ", single_linkage_score, "\n", sep="")
cat("complete_linkage_score = ", complete_linkage_score, "\n", sep="")
cat("ward_linkage_score = ", ward_linkage_score, "\n", sep="")

# The Ward linkage method generated the highest result so we use that

hier_clust_results <- agnes(patients_data_scaled, method = "ward")

# The Ward linkage method generated the highest result so we use that

hc_result <- hclust(dist(patients_data_scaled), "ward.D")
hcdata <- dendro_data(hc_result, type = "rectangle")
ggdendrogram(hcdata, rotate = TRUE, theme_dendro = FALSE) +  labs(title
= "Dendrogram in ggplot2")


# cluster.stats computes Silhouette score, Calinski-Harabasz index, and Dunn index
```

```
hierarchical_clustering_results <-
cluster_analysis(patients_data_scaled, n = k_clusters, method =
"hclust")
hierarchical_clusters <- predict(hierarchical_clustering_results) # the
word "predict" is misleading: the clusters have already been assigned to the data

hier_clusters_stats <- cluster.stats(d=dist(patients_data_scaled),
clustering=hierarchical_clusters, silhouette=TRUE)

cat("The average Silhouette score across the ", k_clusters, " clusters
generated by hierarchical clustering is ",
hier_clusters_stats$avg.silwidth, " (the higher, the better)\n",
sep="")

cat("The Calinski-Harabasz index with ", k_clusters, " clusters
generated by hierarchical clustering is ", hier_clusters_stats$ch, "
(the higher, the better)\n", sep="")

cat("The Dunn index with ", k_clusters, " clusters generated by
hierarchical clustering is ", hier_clusters_stats$dunn, " (the higher,
the better)\n", sep="")

# we compute the Davies-Bouldin index here
davies_bouldin_index_results <- index.DB(x=patients_data_scaled,
cl=hierarchical_clusters)
davies_bouldin_index <- davies_bouldin_index_results$"DB"

cat("The Davies-Bouldin index with ", k_clusters, " clusters generated
by hierarchical clustering is ", davies_bouldin_index, " (the lower,
the better)\n", sep="")

gap_result <- clusGap(patients_data_scaled, FUN = hcut, K.max =
k_clusters, B = number_of_bootstrap_samples, verbose = FALSE)

average_gap <- as.data.frame(gap_result$"Tab")$"gap" %>% mean()

cat("The average Gap statistic across the ", k_clusters, " clusters
generated by hierarchical clustering is ", average_gap, " (the higher,
the better)\n", sep="")
```

F. Repeat all these steps with 3 clusters and then determine which is the better number of clusters between 2 or 3 for hierarchical clustering

G. Determine which is the best method and number of clusters: k-means with 2 clusters, k-means with 3 clusters, hierarchical clustering with 2 clusters, or hierarchical clustering with 3 clusters

H. Insert comments for all the previous R commands you used

Application of the main data cleaning and data preparation steps to a dataset of electronic health records of patients with diabetes type 2 from Saudi Arabia

Repeat all the steps of the previous analysis [A, B, C, ..., I]
Convert the file from XLSX to CSV, first.
Use Diabetic retinopathy (DR) as target for the data unbalance phase

For one-hot encoding, use the `one_hot()` function of the `nestedcv` package:
https://search.r-project.org/CRAN/refmans/nestedcv/html/one_hot.html

AlOlaiwi LA, AlHarbi TJ, Tourkmani AM (2018) Prevalence of cardiovascular autonomic neuropathy and gastroparesis symptoms among patients with type 2 diabetes who attend a primary health care center. PLOS ONE 13(12): e0209500.
https://doi.org/10.1371/journal.pone.0209500

Cerono G, Chicco D. 2024, "Ensemble machine learning reveals key features for diabetes duration from electronic health records". PeerJ Computer Science 10:e1896
https://doi.org/10.7717/peerj-cs.1896

2.2) pone.0209500.s001.xlsx file to download