STA211 : Classification de variables

Vincent Audigier, Ndèye Niang

26 février, 2025

- 1 Objectifs et contexte
- 2 Classification hiérarchique
  - 2.1 Ascendante
  - 2.2 Descendante
- 3 Classification par partitionnement direct
- 4 Exemple
  - 4.1 Chargement des librairies
  - 4.2 Importation des données
  - 4.3 Mise en oeuvre de la classification
    - 4.3.1 CAH
    - 4.3.2 Partitionnement
    - 4.3.3 Comparaison
- 5 Conclusion
- Reférences

1 Objectifs et contexte

Quand les variables sont nombreuses, il est parfois très utile de les regrouper en groupes homogènes. Ceci peut servir à analyser la multicolinéarité entre les variables, ou dans une optique de pré-traitement à réduire leur nombre en leur substituant un petit nombre de variables latentes synthétiques, représentatives de chacune des classes (de variables). Par la suite, on distinguera trois stratégies de partitionnement : les méthodes hiérarchiques ascendantes, descendantes, et les méthodes de partitionnement direct.

2 Classification hiérarchique

2.1 Ascendante

Lors d'une classification ascendante hiérarchique de variables, les groupements se font par agglomération progressive des variables deux à deux. On réunit les variables les plus proches au sens d'une distance à déterminer (ou simplement d'une dissimilarité). Il faut aussi définir un critère d'agrégation afin de pouvoir, par la suite, définir la distance d'une variable ou d'une classe à un groupe déjà établi. Notons que le nombre d'opérations à effectuer pour parvenir à une classification par cette méthode évolue en p^3 où p désigne le nombre de variables à classer. Cela signifie que sur un nombre de variables important, cette méthode ne peut pas être utilisée. Cependant, des algorithmes permettent d'abaisser la complexité en p^2 (voir e.g. Saporta (2006)).

La classification selon cette procédure repose donc sur le choix de l'indice de dissimilarité entre variables et du critère d'agrégation de deux groupes de variables. Les critères d'agrégation utilisés pour la classification de variables sont les mêmes que ceux utilisés pour la classification ascendante hiérarchique des individus, à savoir saut minimal, saut maximal, saut moyen ou méthode de Ward dans le cas où la dissimilarité est une distance euclidienne (i.e. définie à partir d'un produit scalaire). On pourra par exemple consulter Saporta (2006) pour plus de précisions. Nous précisons ici simplement les indices de similarité (ou dissimilarité) les plus courants.

Lorsque que les variables sont de nature **quantitative**, le coefficient de corrélation linéaire est l'indice naturel de similarité utilisé (l'indice de dissimilarité associé s'avère être par ailleurs une distance euclidienne sur les variables).

Lorsque les individus sont décrits par un ensemble de variables **qualitatives**, les indices les plus classiques sont le χ^2 (qui est une distance euclidienne) ou le V de Cramer. Il faut toutefois noter que la nature précise des liaisons entre les variables d'un même groupe sera généralement difficile à apprécier. Il pourra être préférable dans ce cas de se tourner vers la classification des modalités des variables. Pour cela, on peut par exemple considérer le tableau disjonctif associé et effectuer la classification sur les variables indicatrices de ce tableau, ou bien envisager une approche tandem en effectuant une ACM suivie d'une classification à partir des coordonnées des modalités sur les axes (voir Rakotomalala (2025)).

Dans le cas spécifique de variables **binaires** (fréquent lors de l'étude d'associations), on trouve dans Fichet and le Calve (1984) plusieurs indices de similarité définis tous à partir du tableau de contingence des paires de variables $(X_j, X_{j'})$:

- Concordances simples (Sokal, Michener)

$$\frac{N_{11}+N_{00}}{N_{11}+N_{01}+N_{10}}$$

- Jaccard

$$\frac{N_{11}}{N_{11}+N_{01}+N_{10}+N_{00}}$$

- Russel-Rao

$$\frac{N_{11}}{N_{11}+N_{01}+N_{10}}$$

- Ochiai

$$\frac{N_{11}}{\sqrt{(N_{11}+N_{01})(N_{10}+N_{00})}}$$

- Ochiai II

$$\frac{N_{11}\times N_{00}}{\sqrt{(N_{11}+N_{00})(N_{11}+N_{10})(N_{00}+N_{01})(N_{00}+N_{10})}}$$

- Dice

$$\frac{N_{11}}{2N_{11}+N_{01}+N_{10}}$$

- Rogers-Tanimoto

$$\frac{N_{11}+N_{00}}{N_{11}+2(N_{01}+N_{10})+N_{00}}$$

- Kulzinsky

$$\frac{N_{11}}{N_{01}+N_{10}}$$

où

- N_{11} désigne le nombre de caractéristiques communes aux individus i et i' ;
- N_{10} désigne le nombre de caractéristiques possédées par i et pas par i' ;
- N_{01} désigne le nombre de caractéristiques possédées par i' et pas par i ;
- N_{00} désigne le nombre de caractéristiques que ne possèdent ni i, ni i' ;

Une connaissance a priori des données pourrait guider le choix de l'une ou l'autre des mesures. Chacune de ces mesures de similarité possède ses propres propriétés qui influencent les résultats de la classification. On notera que les mesures de dissimilarité associées sont toutes euclidiennes (Fichet and le Calve (1984)).

Enfin, lorsque les données sont **mixtes**, la difficulté essentielle tient à équilibrer les mesures de similarité en fonction de la nature des couples de variables considérés (quantitative/quantitative, quantitative/qualitative ou qualitative/qualitative). On retrouve dans Hummel, Edelmann, and Kopp-Schneider (2017) deux propositions de méthodes ascendantes appelées *CluMix-ama* et *CluMix-dcor*, s'appuyant chacune sur des mesures de similarité spécifiques.

La similarité utilisée dans CluMix-ama pour un couple de variables quantitative est la valeur absolue du coefficient de Spearman. Pour les couples mettant en jeu une (ou deux) variable(s) qualitative(s), les auteurs proposent de définir un ordre qui est relatif à la seconde variable du couple. Par exemple, pour un couple de variables quantitative/qualitative, on calcule la moyenne des rangs de la variable quantitative dans chacune des classes définies par la variable qualitative. L'ordre sur les moyennes définit alors l'ordre des modalités. Une fois cet ordre défini, on calcule la valeur absolue du coefficient γ de Goodman et Kruskal (Agresti (2012), p56-57) qui est une mesure de similarité propre aux couples de variables ordonnées (qu'elles soient quantitatives ou qualitatives). Cette mesure est comprise entre 0 et 1, au même titre que la valeur absolue du coefficient de Spearman utilisée précédemment pour des variables quantitatives.

La similarité utilisée dans la méthode CluMix-dcor est quant à elle construite à partir d'une distance sur les individus. La distance utilisée pour les individus est la distance de Gower (aux facteurs de normalisation près). Soit i et i' deux individus de $\mathbb{R}^p$, cette distance est donnée selon $\sum_{j=1}^{p} d_j(x_{i,j},x_{i',j})$ avec $d_j(x_{i,j},x_{i',j})=|x_{i,j}-x_{i',j}|$ si la variable j est quantitative, $d_j(x_{i,j},x_{i',j})=1_{\{x_{i,j}\neq x_{i',j}\}}$ si la variable j est qualitative, $d_j(x_{i,j},x_{i',j})=|rang(x_{i,j})-rang(x_{i',j})|$ si la variable j est qualitative ordinale. A partir de cette distance entre observations, une mesure de similarité est construite sur les variables (voir Hummel, Edelmann, and Kopp-Schneider (2017) pour plus de précisions).

Ces deux méthodes sont implémentées dans le package R CluMix. Malheureusement, celui-ci n'est plus maintenu. Il est possible de l'installer depuis les archives du CRAN (https://cran.r-project.org/src/contrib/Archive/CluMix/CluMix_2.3.1.tar.gz). Une fois le package téléchargé, le code suivant devrait permettre son installation. Attention à bien modifier le chemin où se situe le dossier téléchargé !

```
install.packages(c("devtools","ClustOfVar","xfun","htmltools","yardstick", "FD","BiocManager"))
devtools::install_github("heike/extracat")
BiocManager::install("marray")
BiocManager::install("Biobase")

#attention, remplacer chemin par ce qu'il convient
install.packages("chemin/CluMix_2.3.1.tar.gz", repos = NULL, type = "source", dependencies = TRUE)
```

Une fois les matrices de (di)similarité définies, il devient assez simple de construire la partition sur le modèle de ce qui a été proposé pour la classification d'individus. Une fois les différentes partitions à K classes, $K-1$ classes, . . . établies, on détermine le niveau de coupure de l'arbre, à partir de l'évolution des hauteurs de fusion.

2.2 Descendante

Les méthodes de classification descendante procèdent par dichotomies successives à la construction d'un arbre hiérarchique descendant dont les segments terminaux constituent une partition des éléments à classer. La partition obtenue est telle que les éléments d'une même classe sont les plus ressemblants possible et deux éléments appartenant à des classes différentes sont les moins ressemblants possible. Un des intérêts majeurs des méthodes descendantes est que l'interprétation des classes obtenues est bien plus aisée car une classe est décrite par les règles de divisions successives.

Une des techniques de classification de variables couramment utilisée est la méthode VARCLUS de SAS (Sarle (1990)) **à ne pas confondre avec la méthode ClustOfVar (Chavent et al. (2012)) définie plus loin**. Elle consiste à :

- réaliser une analyse en composantes principales des variables. Si la seconde valeur propre est supérieure à 1, alors les deux premières composantes factorielles associées aux deux plus grandes valeurs propres sont retenues
- affecter chaque variable à la composante principale avec laquelle elle est le plus corrélée
- répéter les étapes précédentes sur les groupes obtenus tant que la seconde valeur propre est supérieure à 1.

Bien qu'on puisse trouver différentes fonctions R ou packages portant le nom de cette procédure, à ce jour, il ne semble pas en exister d'implémentation sous ce langage. En revanche, on retrouve une implémentation en Python dans la librairie `variable-clustering` par exemple.

3 Classification par partitionnement direct

Parmi les méthodes de partitionnement direct, la méthode la plus connue est la méthode CLV proposée par Vigneau and Qannari (2003) destinée aux variables de nature quantitative. C'est une approche similaire aux K-moyennes, consistant à rechercher une partition en K classes des variables en maximisant un critère exprimant la colinéarité entre les variables d'une classe :

$$n \sum_{k=1}^{K} \sum_{j=1}^{p} \delta_{k,j} cov^2(x_j, u_k)$$

sous la contrainte $u_k u_k^\top = 1$ (pour tout k) où $\delta_{k,j} = 1$ si la variable j est dans la classe k et 0 sinon, et $cov^2(x_j, u_k)$ est le carré de la covariance entre la variable x_j et la variable latente u_k représentant la classe k.

La solution du critère peut être obtenue en utilisant un algorithme de partitionnement alternant deux étapes : l'étape d'affectation des variables aux différents classes (calcul des $\delta_{k,j}$) et l'étape d'estimation des nouveaux centroïdes des classes (calcul des u_k).

Après initialisation des classes, la variable latente u_k pour chaque classe est la première composante principale de l'ACP la matrice dont les variables sont restreintes aux variables de la classe k. Cette composante représente le nouveau centroïde de la classe k. Puis, dans un processus itératif, une variable x_j est affectée à la classe qui maximise le carré de sa covariance avec la variable latente u_k. Ici les groupes de variables constitués sont corrélés positivement ou négativement. Il existe une autre version de la méthode qui tient compte du sens de la liaison de façon à ne pas regrouper des variables opposées. Notons qu'en s'appuyant sur la corrélation comme mesure de similarité, cette approche fait l'hypothèse que les relations entre les variables sont de nature linéaire.

Cette méthode a été généralisée par Chavent et al. (2012) aux variables exclusivement qualitatives ou mixtes en utilisant les composantes principales de l'ACM (pour des données qualitatives) ou de l'AFDM (pour des données mixtes). La méthode proposée porte le nom ClustOfVar. On notera que des versions ascendantes de ces approches, basées sur le même critère d'homogénéité, existent également. Le principe est alors de fusionner les classes de variables de façon à limiter la perte d'homogénéité de la partition. On les retrouvera dans le package *ClustOfVar*.

Une des particularités des méthodes de partitionnement évoquées ici est de pouvoir définir la notion de centroïde d'une classe via les composantes principales d'une analyse factorielle (ACP, ACM ou AFDM en fonction de la nature des variables). Il serait aussi possible d'envisager des approches de types médoïdes (comme les méthodes PAM ou CLARA, voir par exemple Hastie, Tibshirani, and Friedman (2009)) où le centroïde d'une classe est défini à partir d'un élément central de la classe. Dès lors étant donné une matrice de dissimilarité, telles que celles proposées pour la mise en oeuvre des méthodes hiérarchiques, on pourrait définir une partition des variables y compris pour des données mixtes, ou dans un cadre non-linéaire.

4 Exemple

Nous reprenons l'exemple *German Credits* tel qu'abordé dans la section pré-traitement. Nous y appliquons une classification de variables pour données mixtes, d'abord par des approches ascendantes hiérarchiques, puis par des méthodes de partitionnement direct.

4.1 Chargement des librairies

```
library(CluMix)
library(ClustOfVar)
library(cluster)
library(FactoMineR)
```

4.2 Importation des données

```
# importation des données

don <- read.table("https://archive.ics.uci.edu/ml/machine-learning-databases/statlog/german/german.data",
                  sep = " ",
                  stringsAsFactors = TRUE)

# Modification des noms des variables
colnames(don)<-c(
  "Status",
  "Duration",
  "History",
  "Purpose",
  "Credit.Amount",
  "Savings account/bonds",
  "Length.of.current.employment",
  "Instalment.per.cent",
  "Sex.Marital.Status",
  "Guarantors",
  "Duration.in.Current.address",
  "Property",
  "Age.years",
  "Other.installment.plans",
  "Housing",
  "No.of.Credits.at.this.Bank",
  "Job",
  "No.of.dependents"
  ,"Telephone",
 "Foreign.Worker",
 "Creditability")


# Modification des noms des modalités des variables qualitatives
levels(don$Status) <- c("lt.0", "0.to.200", "gt.200", "none_status")

levels(don$History) <-
  c("noCredit.allPaid",
    "thisBank.AllPaid",
    "paidDuly",
    "delay",
    "critical")

levels(don$Purpose) <-
  c(
    "NewCar",
    "UsedCar",
    "Other",
    "Furniture.Equipment",
    "Radio.Television",
    "DomesticAppliance",
    "Repairs",
    "Education",
    "Retraining",
    "Business"
  )

levels(don$`Savings account/bonds`) <-
  c("lt.100", "100.to.500", "500.to.1000", "gt.1000", "Unknown_saving")

levels(don$Length.of.current.employment) <-
  c("lt.1", "1.to.4", "4.to.7", "gt.7", "Unemployed")

levels(don$Sex.Marital.Status) <-
  c(
    "Male.Divorced.Seperated",
```

```
    "Female.NotSingle",
    "Male.Single",
    "Male.Married.Widowed"
  )

levels(don$Guarantors) <- c("None_Guarantor", "CoApplicant", "Guarantor")

levels(don$Property) <-
  c("RealEstate", "Insurance", "CarOther", "Unknown_property")

levels(don$Other.installment.plans) <- c("Bank", "Stores", "None_other_installement")

levels(don$Housing) <- c("Rent", "Own", "ForFree")

levels(don$Job) <-
  c(
    "UnemployedUnskilled",
    "UnskilledResident",
    "SkilledEmployee",
    "Management.SelfEmp.HighlyQualified"
  )

levels(don$Foreign.Worker) <- c("yes_foreign", "no_foreign")

levels(don$Telephone) <- c("none_tel", "yes_tel")

#Codage des variables quantitatives en type "numeric" (plutot que "integer")

##pour la variable Duration
don$Duration <- as.numeric(don$Duration)
class(don$Duration)

## pour les variables Credits.Amount et Age.years
don$Credit.Amount <- as.numeric(don$Credit.Amount)
don$Age.years <- as.numeric(don$Age.years)

#Codage de la variable réponse en type "factor"
don$Creditability <- as.factor(don[, "Creditability"])
levels(don$Creditability) <- c("good", "bad")
```

4.3 Mise en oeuvre de la classification

4.3.1 CAH

4.3.1.1 package *ClustOfVar*

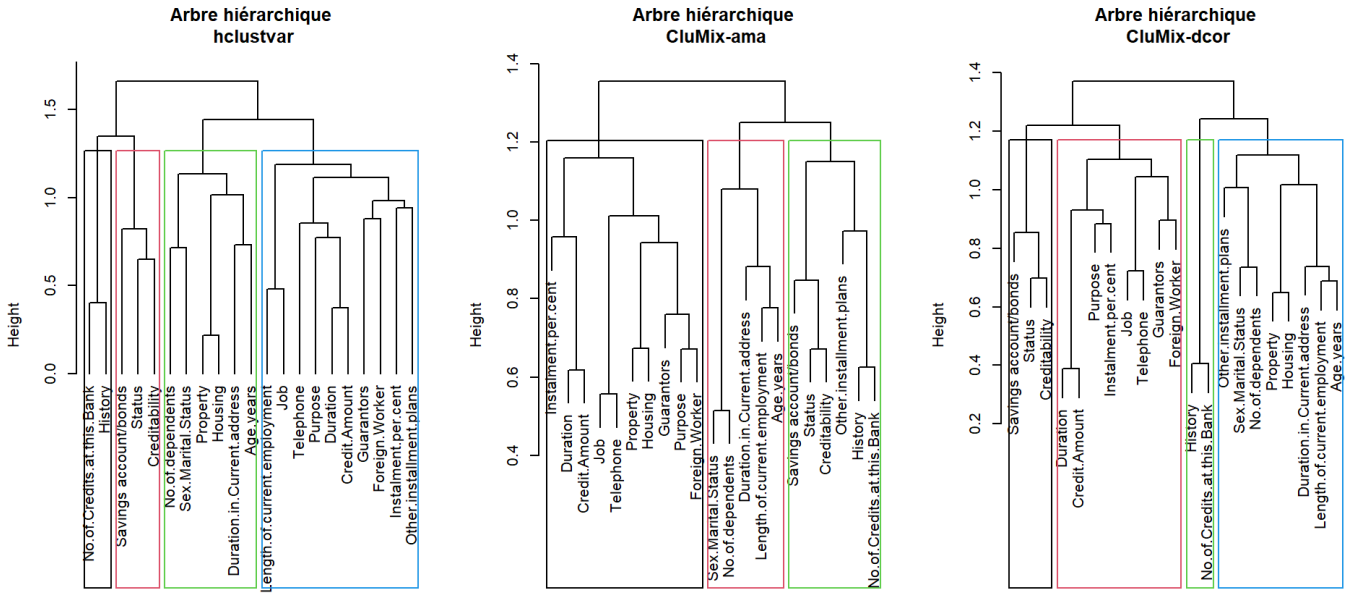
```r
# on fixe la graine du générateur aléatoire pour la reproductibilité des résultats
set.seed(1234)

# on identifie les variables qualitatives et quantitatives
var.factor <- which(sapply(don, class)=="factor")
var.numeric <- which(sapply(don, class)=="numeric"|sapply(don, class)=="integer")
# on effectue la classification
res.cah.hclustvar <- hclustvar(X.quanti = don[,var.numeric],
                               X.quali = don[,var.factor])

# choix du nombre de classes à partir des hauteurs de fusion

# barplot(sort(res.cah.hclustvar$height,decreasing = TRUE),
#         main = "Diagramme des hauteurs de fusion")

# on retient 4 classes
k_hclustvar <- 4

# on récupère la partition associée
res.cutree <- cutreevar(res.cah.hclustvar, k = k_hclustvar)
part.cah.hclustvar <- res.cutree$cluster
```

4.3.1.2 package *CluMix*

```r
## CluMix-ama
res.dendro_ama <- dendro.variables(don)
res.dendro_ama <- as.hclust(res.dendro_ama)

# barplot(sort(res.dendro_ama$height,decreasing = TRUE),
#         main = "Diagramme des hauteurs de fusion")
k_ama <- 3
part.cah.CluMix_ama <- cutree(res.dendro_ama, k = k_ama)

## CluMix-dcor
res.dendro_dcor <- dendro.variables(don, method = "distcor")
res.dendro_dcor <- as.hclust(res.dendro_dcor)

# barplot(sort(res.dendro_dcor$height,decreasing = TRUE),
#         main = "Diagramme des hauteurs de fusion")
k_dcor <- 4
part.cah.CluMix_dcor <- cutree(res.dendro_dcor, k = k_dcor)
```

On visualise les arbres hiérarchiques obtenus

```
par(mfrow=c(1,3))
plot(res.cah.hclustvar,
     main = "Arbre hiérarchique\n hclustvar")
rect.hclust(tree = res.cah.hclustvar,
            k = k_hclustvar,
            border = seq.int(k_hclustvar),
            cluster = part.cah.hclustvar)


plot(res.dendro_ama,
     main = "Arbre hiérarchique\n CluMix-ama",
     xlab = "",
     sub = "")
rect.hclust(tree = res.dendro_ama,
            k = k_ama,
            border = seq.int(k_ama),
            cluster = part.cah.CluMix_ama)

plot(res.dendro_dcor,
     main = "Arbre hiérarchique\n CluMix-dcor",
     xlab = "",
     sub = "")
rect.hclust(tree = res.dendro_dcor,
            k = k_dcor,
            border = seq.int(k_dcor),
            cluster = part.cah.CluMix_dcor)
```



4.3.2 Partitionnement

4.3.2.1 package *ClustOfVar*

```
res.kmeansvar <- kmeansvar(X.quanti = don[,var.numeric],
                           X.quali = don[,var.factor],
                           init = k_hclustvar,
                           nstart = 100)
part.kmeansvar <- res.kmeansvar$cluster
```

4.3.2.2 package *cluster* (algorithme PAM)

```
# selon la dissimilarité issue de CluMix_ama
diss.CluMix_ama <- dist.variables(don)
res.pam_ama <- pam(diss.CluMix_ama, diss = TRUE, k = k_ama)
part.pam_ama <- res.pam_ama$clustering

# selon la dissimilarité issue de CluMix_dcor
diss.CluMix_dcor <- dist.variables(don, method = "distcor")
res.pam_dcor <- pam(diss.CluMix_dcor, diss = TRUE, k = k_dcor)
part.pam_dcor <- res.pam_dcor$clustering
```

4.3.3 Comparaison

Le tableau ci-dessous reporte les différentes partitions obtenues précédemment

```
# on concatène les partitions après les avoir réordonnées selon les noms des variables

tab.mca <- cbind.data.frame(
  cah.clustofvar = as.factor(part.cah.hclustvar[colnames(don)]),
  cah.CluMix_ama = as.factor(part.cah.CluMix_ama[colnames(don)]),
  cah.CluMix_dcor = as.factor(part.cah.CluMix_dcor[colnames(don)]),
  kmeans.clustofvar = as.factor(part.kmeansvar[colnames(don)]),
  pam_ama = as.factor(part.pam_ama[colnames(don)]),
  pam_dcor = as.factor(part.pam_dcor[colnames(don)])
  )


tab.mca
```

Table 4.1: Ensemble des partitions obtenues selon les différentes méthodes.

	cah.clustofvar	cah.CluMix_ama	cah.CluMix_dcor	kmeans.clustofvar	pam_ama	pam_dcor
Status	4	1	1	3	1	1
Duration	1	2	2	4	2	2
History	3	1	3	2	1	3
Purpose	1	2	2	4	2	2
Credit.Amount	1	2	2	4	2	2
Savings account/bonds	4	1	1	3	2	1
Length.of.current.employment	1	3	4	1	3	4
Instalment.per.cent	1	2	2	1	2	2
Sex.Marital.Status	2	3	4	2	3	4
Guarantors	1	2	2	3	2	1
Duration.in.Current.address	2	3	4	1	2	4
Property	2	2	4	1	2	2
Age.years	2	3	4	1	3	4
Other.installment.plans	1	1	4	4	1	3
Housing	2	2	4	1	3	4
No.of.Credits.at.this.Bank	3	1	3	2	1	3
Job	1	2	2	4	2	2
No.of.dependents	2	3	4	2	3	4
Telephone	1	2	2	4	2	2
Foreign.Worker	1	2	2	4	2	2

	cah.clustofvar	cah.CluMix_ama	cah.CluMix_dcor	kmeans.clustofvar	pam_ama	pam_dcor
Creditability	4	1	1	3	2	1

Afin de comparer ces différentes partitions, on effectue une ACM sur ce tableau constitué des variables en lignes et les partitions en colonnes.

```
MCA(tab.mca)
```

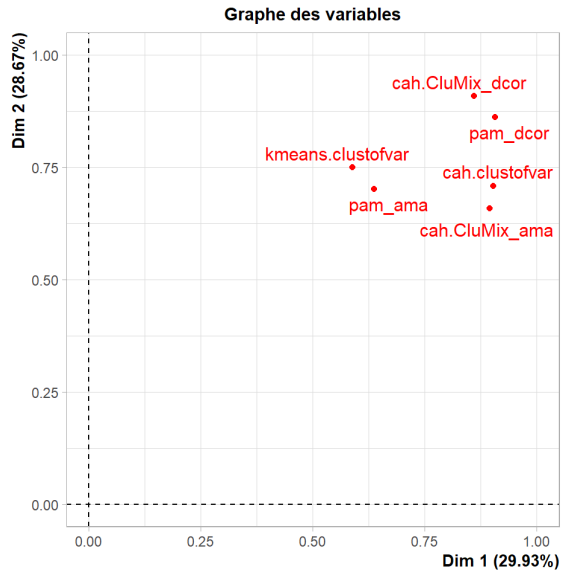


Figure 4.1: Graphe des variables sur le plan principal.

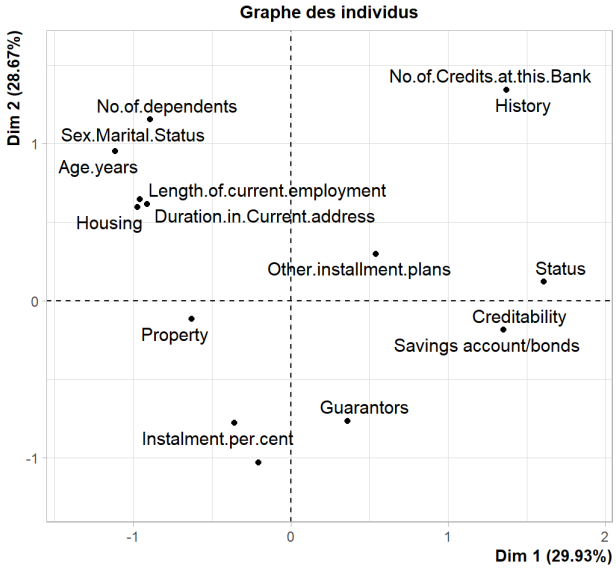


Figure 4.2: Graphe des individus sur le plan principal.

Table 4.2: Coordonnées, contribution et cosinus carrés pour les deux premiers axes. Les variables sont ordonnées selon la qualité de représentation sur le premier plan.

	coord Dim 1	coord Dim 2	contrib Dim 1	contrib Dim 2	cos2 Dim 1	cos2 Dim 2
Age.years	-1.11	0.95	7.41	5.62	0.48	0.35
Purpose	-0.20	-1.03	0.25	6.61	0.03	0.78
Credit.Amount	-0.20	-1.03	0.25	6.61	0.03	0.78
Job	-0.20	-1.03	0.25	6.61	0.03	0.78
Telephone	-0.20	-1.03	0.25	6.61	0.03	0.78
Foreign.Worker	-0.20	-1.03	0.25	6.61	0.03	0.78
Duration	-0.20	-1.03	0.25	6.61	0.03	0.78

	coord Dim 1	coord Dim 2	contrib Dim 1	contrib Dim 2	cos2 Dim 1	cos2 Dim 2
Sex.Marital.Status	-0.89	1.15	4.75	8.27	0.28	0.46
No.of.dependents	-0.89	1.15	4.75	8.27	0.28	0.46
History	1.37	1.34	11.24	11.21	0.31	0.30
No.of.Credits.at.this.Bank	1.37	1.34	11.24	11.21	0.31	0.30
Housing	-0.97	0.60	5.67	2.21	0.42	0.16
Status	1.61	0.12	15.45	0.09	0.57	0.00
Length.of.current.employment	-0.96	0.64	5.50	2.58	0.39	0.18
Duration.in.Current.address	-0.91	0.62	4.98	2.36	0.38	0.17
Instalment.per.cent	-0.36	-0.78	0.76	3.77	0.09	0.42
Savings account/bonds	1.35	-0.19	10.92	0.21	0.46	0.01
Creditability	1.35	-0.19	10.92	0.21	0.46	0.01
Guarantors	0.36	-0.76	0.78	3.64	0.06	0.27
Property	-0.63	-0.12	2.38	0.09	0.24	0.01
Other.installment.plans	0.54	0.30	1.75	0.55	0.10	0.03

Le graphe des variables permet d'identifier une relative proximité entre les différentes partitions. Notons que le graphe des individus fournit une visualisation des proximités entre les variables assez intéressante permettant d'apprécier les groupes de variables qui se dégagent de ces différentes partitions :

- Un premier groupe (en haut à gauche) composé des variables `Length.of.current.employment, Sex.Marital.Status, Duration.in.Current.address, Age.years, Housing, No.of.dependents`. On retrouve ici des variables caractérisant le profil personnel du demandeur
- un deuxième (en bas) composé des variables `Duration, Purpose, Credit.Amount, Instalment.per.cent, Guarantors, Job, Telephone, Foreign.Worker` (partiellement visible sur la figure en raison de la juxtaposition des labels). On retrouve ici des variables ayant trait au type de crédit demandé (Duration, Purpose, Credit.Amount) d'autres liées au profil financier du demandeur (Instalment.per.cent, Guarantors)
- un troisième (à droite) composé des variables `Status, Savings account/bonds, Creditability`, directement reliées aux actifs du demandeur.
- un quatrième groupe (en haut à droite) constitué des variables `History, No.of.Credits.at.this.Bank`

On constatera que la variable *Creditability* est plutôt positionnée dans le troisième groupe, ce qui suggère qu'elle est liée de façon privilégiée avec les variables de ce groupe et peu avec celles des autres groupes.

5 Conclusion

La classification de variables vise à regrouper les variables en classes homogènes : les variables dans un même groupe sont fortement liées entre elles, les variables dans des classes différentes sont faiblement liées. Comme pour la classification des individus, on peut procéder par des méthodes de partitionnement hiérarchiques ou directes.

Cette étape de classification est parfois au coeur de certaines techniques de fouille. On pense notamment à la méthode des *fuzzy forest* (Conn et al. (2019)), une variante des forêts aléatoires en présence de variables corrélées. Le principe de cette méthode est d'effectuer dans un premier temps une classification des variables, afin de déterminer des groupes de variables corrélées entre elles. Ensuite, une sélection de variables de type *backward* est effectuée au sein de chaque groupe sur la base des mesures d'importance données par une forêt. Par la suite, cette procédure de sélection est répétée sur l'ensemble des variables sélectionnées de chaque groupe. Cette procédure permet notamment d'obtenir in fine une liste ordonnée des mesures d'importance qui peut être utilisée pour construire un prédicteur à partir d'un sous-ensemble de variables. On retrouvera une implémentation de cette méthode dans le package R `fuzzyforest`.

Il est fréquent de vouloir effectuer à la fois de la classification des individus et des variables sur un même jeu de données. Dès lors, on peut se demander si certains groupes d'individus ne seraient pas caractérisés par un seul des groupes de variables plutôt que par l'intégralité de celles-ci. Ceci est fréquent dans certains domaines comme la bioinformatique, le textminig, le webmining ou encore le marketing : des clients peuvent avoir des habitudes de consommation complètement différentes par rapport à certains produits, par exemple, une personne ayant un nourrisson achètera fréquemment des couches et du lait maternel, tandis qu'une personne sans enfant n'en achètera pas. Ces clients sont donc complètement opposés par rapport à ces produits (variables). Pour

autant s'ils sont tous les deux amateurs de cuisine asiatique ils achèteront fréquemment des pâtes de riz et de la sauce soja. En effectuant uniquement de la classification des consommateurs, ces deux individus ont peu de chances d'être dans une même classe, alors qu'ils ont pourtant certaines habitudes de consommation communes. Pour prendre en compte cet aspect, on peut alors procéder par *biclustering* (aussi appelé *co-clustering* ou *classification croisée*). Le principe est de constituer simultanément des groupes d'individus et de variables tels que chaque groupe d'individus ne soit construit qu'à partir des ressemblances vis-à-vis de certaines variables d'un même groupe. Il existe différentes façons d'effectuer cette classification (voir Charrad and Ben Ahmed (2011)).

Références

Agresti, Alan. 2012. *Categorical Data Analysis, 3rd Edition*. Hoboken, NJ, USA: Wiley.

Charrad, Malika, and Mohamed Ben Ahmed. 2011. "Simultaneous Clustering : a survey." In *4th International Conference on Pattern Recognition and Machine Intelligence. PReMI 2011*, LNCS 6744:370–75. Springer. Moscow, Russia. https://hal.archives-ouvertes.fr/hal-01125890 (https://hal.archives-ouvertes.fr/hal-01125890).

Chavent, Marie, Vanessa Kuentz-Simonet, Benoît Liquet, and Jérôme Saracco. 2012. "ClustOfVar: An r Package for the Clustering of Variables." *Journal of Statistical Software, Articles* 50 (13): 1–16. https://doi.org/10.18637/jss.v050.i13 (https://doi.org/10.18637/jss.v050.i13).

Conn, Daniel, Tuck Ngun, Gang Li, and Christina M. Ramirez. 2019. "Fuzzy Forests: Extending Random Forest Feature Selection for Correlated, High-Dimensional Data." *Journal of Statistical Software* 91 (9): 1–25. https://doi.org/10.18637/jss.v091.i09 (https://doi.org/10.18637/jss.v091.i09).

Fichet, B., and G. le Calve. 1984. "Structure géométrique des principaux indices de dissimilarité sur signes de présence-absence." *Stat. Anal. Données* 9 (3): 11–44.

Hastie, Trevor, Robert Tibshirani, and Jerome Friedman. 2009. *The Elements of Statistical Learning*. 2nd ed. Springer Series in Statistics. New York, NY, USA: Springer New York Inc. https://hastie.su.domains/ElemStatLearn/printings/ESLII_print12_toc.pdf.download.html (https://hastie.su.domains/ElemStatLearn/printings/ESLII_print12_toc.pdf.download.html).

Hummel, Manuela, Dominic Edelmann, and Annette Kopp-Schneider. 2017. "Clustering of Samples and Variables with Mixed-Type Data." *PLOS ONE* 12 (11): 1–23. https://doi.org/10.1371/journal.pone.0188274 (https://doi.org/10.1371/journal.pone.0188274).

Rakotomalala, Ricco. 2025. "Classification Des Variables Qualitatives." https://eric.univ-lyon2.fr/ricco/cours/slides/classif_variables_quali.pdf (https://eric.univ-lyon2.fr/ricco/cours/slides/classif_variables_quali.pdf).

Saporta, G. 2006. *Probabilités, Analyse Des Données Et Statistique*. Editions Technip.

Sarle, WS. 1990. "The Varclus Procedure. Sas/Stat User's Guide. Sas Institute." *Inc., Cary, NC, USA*.

Vigneau, Evelyne, and EM Qannari. 2003. "Clustering of Variables Around Latent Components." *Communications in Statistics-Simulation and Computation* 32 (4): 1131–50.