Classification de variables et analyse multivariée de données mixtes issues d'une étude BCI

Jérôme Saracco, Marie Chavent, Liliana Garcia-Audin, Véronique Lespinet-Najib, Ricardo Ron-Langevin

Classification de variables et analyse multivariée de données mixtes issues d'une étude BCI

Clustering of variables and multivariate analysis of mixed data from BCI study

Jérôme Saracco[1], Marie Chavent[2], Liliana Garcia[3], Véronique Lespinet-Najib[4], Ricardo Ron-Angevin[5]

[1] IMB UMR 5251 & CQFD Inria Bordeaux Sud Ouest, ENSC - Bordeaux INP, Bordeaux, France, jerome.saracco@ensc.fr
[2] IMB UMR 5251 & CQFD Inria Bordeaux Sud Ouest, Université de Bordeaux, Bordeaux, France, marie.chavent@u-bordeaux.fr
[3] IMS UMR 5218, CIH, CNRS - Université de Bordeaux, Bordeaux, France, liliana.garcia@ensc.fr
[4] IMS UMR 5218, CIH, ENSC - Bordeaux INP, Bordeaux, France, veronique.lespinet@ensc.fr
[5] Departamento de Tecnologìa Electrónica, Universidad de Málaga, Campus de Teatinos, Málaga, Espagne, rra@dte.uma.es

RÉSUMÉ. L'objectif de ce travail est de traiter des données complexes issues de la technique des *Brain Computer Interfaces* (BCI) au moyen de méthodes statistiques multivariées (approche PCAmix et classification de variables) afin de mieux comprendre et interpréter les relations qui existent entre elles. Cet article présente ainsi la classification de variables qui a pour but de réunir des variables fortement liées entre elles. L'approche proposée fonctionne avec des données mixtes, c'est à dire des données contenant des variables numériques et des variables catégorielles. Deux algorithmes de classification de variables sont décrits : un de classification hiérarchique et un autre de partitionnement de type k-means. Une rapide description de la méthode PCAmix (qui permet de faire de l'analyse en composantes principales pour des données mixtes) est fournie, vu que le calcul des variables synthétiques résumant les classes de variables obtenues est fondé sur cette méthode multivariée. Enfin, les approches PCAmix et ClustOfVar (implémentées dans les packages R **ClustOfVar** et **PCAmixdata**) sont mises en œuvre sur les données réelles issues de l'étude BCI. Des recommandations, reposant non seulement sur des critères de performances, d'efficience mais aussi de satisfaction, ont pu être faites concernant le choix d'interface dans l'usage des claviers virtuels, notamment pour des personnes avec des problèmes moteurs tels que la maladie de Charcot.

ABSTRACT. The aim of this work is to analyze complex data from a Brain Computer Interfaces (BCI) study using multivariate statistical methods (PCAmix approach and clustering of variables) to better understand and interpret their relationships. This article presents clustering of variables which aim is to lump together strongly related variables. The proposed approach works on a mixed data set, i.e. on a data set which contains numerical variables and categorical variables. Two algorithms of clustering of variables are described : a hierarchical clustering and a k-means type clustering. A brief description of PCAmix method (that is a principal component analysis for mixed data) is provided, since the calculus of the synthetic variables summarizing the obtained clusters of variables is based on this multivariate method. Finally, the PCAmix and ClustOfVar approaches (implemented in the R packages **ClustOfVar** and **PCAmixdata**) are illustrated on a real dataset from a BCI (brain computer interface) study. Recommendations, based not only on performance, efficiency, but also on satisfaction criteria, could be made concerning the choice of interface in the use of virtual keyboards, especially for people with motor disorder such as Charcot's disease.

MOTS-CLÉS. Classification de variables, données mixtes, analyse en composantes principales, packages R, brain computer interface, analyse multivariée des données, visualisation des données.

KEYWORDS. clustering of variables, mixed data, principal component analysis for mixed data, R packages, brain computer interface, multivariate data analysis, data visualization.

1. Introduction et problématique de l'étude BCI

La technique BCI (Brain Computer Interface) est une technique sophistiquée qui utilise l'activité cérébrale pour commander des appareils externes (interface) sans passer par une action des muscles squelettiques. Une étude a été réalisée pour commander un clavier virtuel par BCI, ceci afin de communiquer de façon non verbale. Ce type d'interfaces peut être d'une grande utilité pour des personnes ayant des

problèmes moteurs tels que la scélorose amyotrophique latérale. L'objet de cette étude BCI est d'optimiser l'acceptabilité de ce type d'interfaces qui apparait comme étant complexe et difficile à utiliser. A cette fin, des mesures objectives et subjectives ont été effectuées pour étudier l'effet de la taille du clavier sur la performance, la charge de travail et l'utilisabilité de l'interface. Les quantités mesurées (appelées variables en statistique) sont aussi bien quantitatives (tels que des pourcentages d'erreurs, ou encore des scores prenant des valeurs entre 0 et 10) que catégorielles (comme la taille de l'écran qui peut être petit, moyen ou grand, ou encore le niveau de stress prenant les mêmes modalités). On parle de données mixtes (mélange de variables quantitatives et de variables catégorielles mesurées sur les mêmes individus de l'étude). Les détails sur cette étude BCI et son protocole, ainsi que sur les diverses variables disponibles, sont fournis aux sections 1.2. et 1.3.

L'objectif de cet article est de faire une présentation de deux approches de statistique descriptive multidimensionnelle pour données mixtes permettant d'obtenir une description synthétique et une visualisation des données de cette étude BCI, et en particulier des liens existant entre les diverses variables de l'étude. Les deux méthodes mises en œuvre ici sont l'analyse en composantes principales pour données mixtes (appelée PCAmix ci-après) et la classification de variables pour données mixtes (appelée ClusOfVar ci-après). Une brève introduction sur ces approches statistiques est donnée à la section 1.1.

Dans la suite de cet article, les principes des approches PCAmix et ClustOfVar sont détaillés et illustrées sur les données de l'étude BCI considérée. La section 2. est dédiée à la classification de variables pour données mixtes. Dans la section 3., l'analyse en composantes principales pour des données mixtes est présentée. Notons que la méthode PCAmix est la méthode sous-jacente permettant la construction des variables synthétiques des classes de l'approche ClustOfVar. La section 4. est dédiée à l'illustration des méthodes de classification de variables et de la méthode PCAmix sur le jeu de données réelles issu de l'étude BCI en utilisant les packages R **PCAmixdata** et **ClustOfVar**.

1.1. *Introduction à l'analyse en composantes principales (PCAmix) et à la classification de variables (ClusOfVar) pour données mixtes*

L'objectif de la **classification d'individus** (observations) est de réunir des observations similaires, c'est à dire de trouver des groupes (*clusters* en anglais) dans les données. Plus précisément, il existe des méthodes et des algorithmes associés permettant de séparer n individus en K groupes d'individus. Chaque groupe peut alors être résumé par le biais d'un représentant (individu moyen par exemple). Ceci peut être vu comme une étape de réduction de dimension dans le sens où l'on passe de n individus à K individus (virtuels) représentant leurs groupes respectifs. Pour des *données numériques* (ou quantitatives), c'est à dire lorsque les individus sont uniquement caractérisés par des variables numériques, il existe des méthodes standards de classification. On peut citer par exemple la classification hiérarchique (donnant lieu à un dendrogramme ou arbre de classification) ou la classification de type *k-means*. Pour des *données catégorielles* (ou qualitatives), une première étape consiste à appliquer la méthode d'analyse factorielle multiple des correspondances sur les données d'origine afin d'obtenir des composantes principales (aussi appelées scores) qui sont de nouvelles variables synthétiques numériques résumant l'information du jeu de données. Dans une seconde étape, les méthodes standards de classification peuvent être appliquées afin de déterminer des groupes d'individus dans les données. Pour des *données mixtes* (c'est à dire quand les n individus sont décrits par un mélange de variables numériques et catégorielles), une manière de

faire des groupes d'individus est de considérer à nouveau deux étapes. La première consiste à appliquer la méthode PCAmix (qui une méthode d'analyse en composantes principales pour données mixtes) sur les données d'origine. Puis, dans une seconde étape, comme précédemment, des méthodes standards de classification d'individus peuvent être mises en œuvre sur les composantes principales issues de PCAmix (qui sont de nouvelles variables synthétiques numériques).

Dans cet article, nous nous focalisons sur la **classification de variables**, ces dernières peuvent être numériques et/ou catégorielles. Le but de la classification de variables est de grouper ensemble les variables fortement liées entre elles, c'est à dire de séparer les variables en classes (de variables). Il sera possible de résumer chaque classe de variables par une seule variable synthétique numérique. Cette approche peut être vue comme une étape de réduction de dimension dans le sens où les p variables initiales peuvent être résumées par K nouvelles variables synthétiques. Ainsi, la classification de variables apparaît comme une bonne alternative à l'analyse en composantes principales pour données mixtes (PCAmix). En effet, elle permet de supprimer les redondances d'information dans les p variables disponibles. Notons que, pour chaque groupe de variables, la variable synthétique résumant l'information apportée par les variables du groupe sera construite uniquement avec les variables du groupe, contrairement à l'analyse en composantes principales où les variables synthétiques sont construites avec l'ensemble des p variables. L'approche par classification de variables permet ainsi d'interpréter plus facilement les variables synthétiques.

Pour des données purement numériques, des approches de classification de variables sont disponibles dans la littérature statistique. Citons par exemple : *likelihood linkage analysis* (voir Lerman [1991]) qui est une approche de type "model-based", *diametrical clustering* (voir Dhillon et al. [2003]), *clustering around latent variables* (voir Vigneau et Qannari [2003] ; Vigneau et al. [2015]). Pour les données mixtes, Chavent et al. [2012a] ont introduit l'approche spécifique de classification de variables appelée ClustOfVar. Ce approche pouvant traiter les données mixtes, il peut bien évidemment également traiter le cas de données purement numériques ou purement catégorielles.

1.2. *Problématique de l'étude BCI*

Les systèmes de Brain Computer Interfaces (BCI) sont un domaine fascinant de recherche dans le cadre des études sur les systèmes d'aides et de substitution. Les BCI doivent permettre d'effectuer des tâches à l'aide du contrôle mental. Cette technologie est basée sur un système mesurant et analysant l'activité cérébrale en temps réel (cf. Figure 1) par l'intermédiaire d'enregistrements électro-encéphalographiques. L'analyse des modifications en temps réel des activités cérébrales, associée avec des signaux de contrôle, permet de transmettre des ordres aux dispositifs associés (ordinateurs, fauteuil électrique, etc.).

Par exemple, cette technologie novatrice peut aider des personnes ayant perdu n'importe quelle capacité motrice pour interagir ou communiquer avec son environnement. Ce type d'interfaces peut-être d'une grande utilité pour des personnes avec des problèmes moteurs tels que la sclérose latérale amyotrophie (SLA ou maladie de Charcot). Il existe plusieurs systèmes BCI, un des systèmes classiques (notamment pour des dispositifs d'aides à la communication) repose sur les potentiels évoqués P300. Le signal P300, enregistré sur les régions centrale et pariétale, est une déviation positive des ondes cérébrales à une latence d'environ 300 ms après la présentation du stimulus. Dans ce type de dispositif, une matrice de 6×6 caractères, disposés en rangées et en colonnes, est montrée au sujet. Les lignes et les colonnes

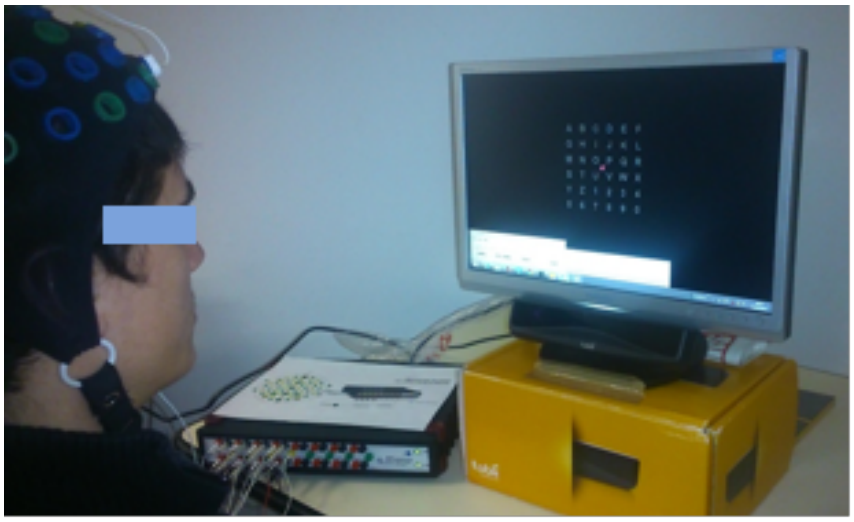

Figure 1. Système de commande de clavier virtuel contrôlé à travers une Interface Cerveau-ordinateur (*BCI - Brain Computer Interface*). Le sujet est assis face à l'écran pour effectuer la tâche d'écriture. Les signaux EEG sont captés par les électrodes et amplifiés avant d'être traités par l'ordinateur utilisant le logiciel BCI-2000. Les lettres "flashent" de manière aléatoire et le sujet doit concentrer son attention sur le symbole cible (lettre ou chiffre) pour produire le signal de commande nommée P300. *Photo courtoisie ENSC.*

sont flashées de manière aléatoire, l'une après l'autre, tandis que l'utilisateur concentre son attention sur l'élément matriciel qu'il souhaite sélectionner (comme un symbole ou une lettre). Après un certain nombre de flash, l'ordinateur identifie l'élément de la matrice "pensé" par l'utilisateur comme l'intersection de la rangée et de la colonne provoquant un P300 plus grand, et ce symbole est présenté sur l'écran. L'efficacité du système d'épellation BCI basé sur le P300 est garantie par un certain nombre d'études réalisées non seulement sur des sujets sains mais aussi sur des sujets atteints d'une incapacité motrice, voir Marchetti et Priftis [2014]. Dans l'ensemble, ces études concluent que le BCI basé sur le P300 est un outil de communication efficace pour les personnes qui ont perdu ou qui sont en train de perdre leur capacité d'écrire ou de parler. Cependant, il est encore nécessaire d'améliorer la facilité d'utilisation de ces systèmes d'épellation car les temps de réalisation d'une tâche d'épellation peuvent être encore longs et avec des erreurs. L'utilisabilité, en conception centrée utilisateur et en cognitique, est un concept majeur qui repose sur 3 dimensions : l'efficacité, l'efficience et la satisfaction d'un sujet en interaction avec une interface. La définition ISO actuelle de l'utilisabilité [ISO 9241-11] comprend trois mesures : efficacité (exactitude et exhaustivité avec lesquelles les utilisateurs atteignent les objectifs fixés), efficience (ressources déployées pour atteindre les objectifs) et satisfaction (ressenti subjectif des utilisateurs), voir Nielsen [1993] et Nielsen [1994]. Certains facteurs, tels que la fatigue mentale induite par un usage prolongé (voir Kececi et al. [2006] et Murata et Uetake [2001]), une attention soutenue à un symbole sur l'écran (voir Mangun et Buck [1998]), la motivation de l'utilisateur (voir Kleih et al. [2010] ou Kececi et al. [2006]) ou une frustration due à une erreur, peuvent influencer l'amplitude et la latence du composant P300 (voir Polich et Kok [1995] pour une revue). À cet égard, l'influence sur la performance des aspects temporels et spatiaux des interfaces utilisateurs de ces systèmes attire de plus en plus l'attention des chercheurs, voir Schalk et al. [2004].

Bien que la grande majorité des articles se soit concentrée sur les algorithmes de traitement du signal afin d'améliorer les performances du système P300-BCI, plusieurs recherches ont étudié des paramètres susceptibles d'influencer les performances de l'utilisateur. Certains de ces paramètres sont les caractéristiques de synchronisation du stimulus (voir Lu et al. [2013] ou McFarland et al. [2011]), l'effet du

contraste de luminosité (voir Li et al. [2014]) et l'influence du contraste de couleur de l'interface (voir Nam et al. [2010]). En ce qui concerne l'effet de la configuration de la matrice, la recherche est limitée. Certaines études ont montré comment la taille de la matrice affecte les performances des tâches de l'utilisateur. Allison et Pineda [2003] ont mené une étude comparant trois tailles de matrice différentes (4×4, 8×8, et 12×12). Les résultats ont indiqué que les matrices plus grandes évoquaient une plus grande amplitude de P300 et que la taille de la matrice n'avait pas d'effet significatif sur les performances ou les préférences. D'autre part, une étude comparant deux tailles matricielles différentes a conclu que la précision était plus élevée pour la matrice 3×3, alors que l'amplitude P300 était plus élevée pour la matrice 6×6, voir Sellers et al. [2006]. Dans les deux études, la taille des symboles était la même dans les différentes matrices, et ainsi les plus grandes matrices apparaissaient plus grandes sur l'écran.

L'objectif principal de l'étude présentée ici est d'évaluer la performance de BCI en utilisant trois tailles d'écrans spécifiques, afin de mesurer l'impact de la taille du clavier sur l'utilisabilité de l'interface. De plus, il nous semblait important d'étudier 2 types de conditions de contrainte en rapport avec la mobilité (ou non) des yeux ; que nous appellerons ici "condition fixe" vs "condition mobile" respectivement.

Les contraintes des analyses des données issues des études BCI sont multiples : nombre de sujets souvent réduit dû à la durée des expériences (2 heures sur 4 jours différents), données multivariées (nombreuses variables numériques et/ou catégorielles), difficulté pour représenter et visualiser les relations entre les variables au niveau de l'ensemble des données. Classiquement, la majorité des études repose sur des analyses de type "statistique inférentielle" (tests de comparaison de moyennes notamment avec des analyses de variance par exemple) ne prenant en compte qu'une seule variable à expliquer par un ou plusieurs facteurs (par exemple ici la taille de l'écran et/ou la condition fixe ou mobile). Le travail présenté dans cet article a pour but de mettre en évidence l'intérêt d'utiliser des approches d'analyse de données multidimensionnelles, notamment la classification de variables (ClustOfVar) et et l'analyse en composantes principales pour données mixtes (PCAmix) afin d'avoir une vision plus globale des liens existant entre les différentes variables, qu'elles soient numériques ou catégorielles.

1.3. *Description des données*

1.3.1. Description des sujets

Un total de 12 sujets (sept hommes et cinq femmes, âgés de 19-25 ans, moyenne = 20,6 $\pm$ 0,9 ans) a participé à la présente étude, qui comprenait 6 séances, une pour chaque taille du clavier (petit, intermédiaire et grand) et pour chaque condition (fixe et mobile) Selon les auto-évaluations, tous les participants n'avaient pas d'antécédent de maladie neurologique ou psychiatrique, avaient une vision normale, et ont donné leur consentement éclairé pour leur participation à l'étude. Aucun n'avait d'expérience antérieure avec les systèmes BCI.

1.3.2. Protocole

Nous avons réalisé des expériences sur des sujets sains soumis à deux types de conditions de contrainte en rapport avec la mobilité (ou non) des yeux : ici "condition mobile" vs "condition fixe" respectivement. Les sujets passent trois sessions à quelques jours d'intervalle dans lesquels ils vont tester trois tailles de

clavier (petit, intermédiaire ou grand) dans les deux conditions mobile et fixe décrites précédemment. Les sujets, assis à une distance de 60 cm par rapport à l'écran, doivent focaliser leur attention sur les symboles (flashs) correspondant à l'écriture de trois "mots" prédéterminés : CHAT, PURE, 1935.

Un pré-calibrage est effectué au début de l'expérience pour déterminer si l'individu est capable de maitriser bien le système et arriver au 100% de performance BCI. Nous déterminons alors le nombre de flashs nécessaires pour arriver au maximum de calibration.

Les symboles présentés (lettres ou chiffres) ont été choisis par rapport à leur localisation et à leur distribution dans trois zones désignées (jaune, vert, rouge) selon la distance par rapport au centre et dont la couleur est représentée à la figure **2**.

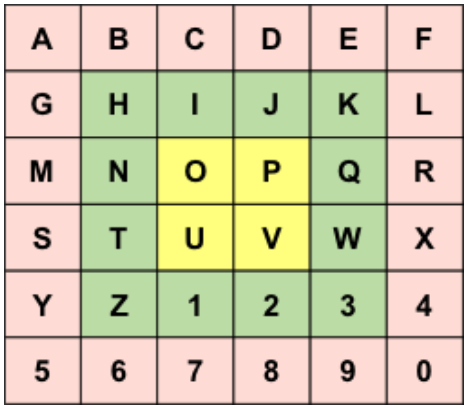

Figure 2. Exemple d'une matrice 6×6 de symboles (lettres ou chiffres), les couleurs indiquent la distance par rapport au centre.

1.3.3. Description des variables disponibles

Comme mentionné dans l'introduction, l'objectif principal de cette étude était d'évaluer la facilité d'utilisation de 3 tailles de clavier dans deux conditions de contraintes de mouvements des yeux (fixe vs mobile), et selon les trois dimensions de l'utilisabilité : efficacité, efficience et satisfaction.

Voici la liste des $p = 25$ variables considérées dont $p_1 = 17$ sont des variables numériques et $p_2 = 8$ sont des variables catégorielles.

Les deux variables concernant le cadre expérimental sont :

- `Taille` : variable catégorielle à 3 modalités `P` (petit) , `I` (intermédiaire), `G` (grand),
- `Condition` : variable catégorielle à 2 modalités `fixe` et `mobile`.

Pour la mesure de la dimension "efficacité", 6 variables numériques ont été mesurées :

- `Pct.Erreur` : pourcentage global d'erreurs,
- `Pct.errZONE.ROUGE` : pourcentage d'erreurs dans la zone rouge,
- `Pct.errZONE.JAUNE` : pourcentage d'erreurs dans la zone jaune,
- `Pct.errZONE.VERT` : pourcentage d'erreurs dans la zone verte,
- `Nb.flash.max.cal` : nombre de flashs nécessaires au sujet pour arriver au maximum de calibration,
- `max.Pct.cal` : pourcentage maximum de réussite obtenu par le sujet lors du calibrage du système.

Pour la mesure de la dimension "efficience", 11 variables numériques ont été mesurées :

- `FATIGUE.VAS` : score de fatigue (VAS pour *Visual Analogue Scale*), noté sur une échelle de 0 (fatigue faible) à 10 (fatigue forte),

ainsi que différents scores de charge mentale (évaluée par le test NASA-TLX) :

- `Exigence.mentale`, score noté sur une échelle de 0 (faible) à 10 (forte),
- `Exigence.temporelle`, score noté sur une échelle de 0 (faible) à 10 (forte),
- `Degree.Performance`, score noté sur une échelle de 0 (faible) à 10 (forte),
- `Exigence.physique`, score noté sur une échelle de 0 (faible) à 10 (forte),
- `Effort`, score noté sur une échelle de 0 (faible) à 10 (forte),
- `Frustration`, score noté sur une échelle de 0 (faible) à 10 (forte),
- `WKL.total`, score calculé selon une formule tenant compte de l'ensemble des scores précédents,

et différentes mesures de facilité de perception :

- `PERC1.symboles`, mesure de la difficulté à percevoir les différents symboles, notée sur une échelle de 0 (faible) à 10 (forte),
- `PERC2.Distance`, mesure de la difficulté à percevoir les symboles éloignés du centre de l'écran, notée sur une échelle de 0 (faible) à 10 (forte),
- `PERC3.lig.col`, mesure de la difficulté à percevoir les différentes lignes et colonnes, notée sur une échelle de 0 (faible) à 10 (forte).

Pour la mesure de la dimension "satisfaction", 6 variables catégorielles à trois modalités ont été utilisées (les modalités sont ordonnées de 1 à 3, le niveau 1 correspondant à "faible", le niveau 2 à "moyen", le niveau 3 à "élevé") :

- `niv.complex` : usage complexe (modalités : `Cplx1`, `Cplx2`, `Cplx3`),
- `niv.confort` : usage confortable (modalités : `Conf1`, `Conf2`, `Conf3`),
- `niv.stress` : usage stressant (modalités : `Stre1`, `Stre2`, `Stre3`),
- `niv.fatig` : usage fatiguant (modalités : `Fati1`, `Fati2`, `Fati3`),
- `niv.ctrl` : usage controlable (modalités : `Ctrl1`, `Ctrl2`, `Ctrl3`),
- `niv.pref` : niveau de préférence globale (modalités : `Pref1`, `Pref2`, `Pref3`).

2. La classification de variables pour données mixtes : l'approche ClusOfVar

Soit $P_K = (C_1, \ldots, C_k, \ldots, C_K)$ une partition en K classes des p variables observées (p_1 étant numériques, p_2 étant catégorielles, avec $p_1 + p_2 = p$) où C_k est la $k^{\text{ème}}$ classe des variables. Une variable $\mathbf{x}_j$ appartenant à la classe C_k peut être numérique ou catégorielle :

- $\mathbf{x}_j \in \mathbb{R}^n$ quand la variable j est numérique,
- $\mathbf{x}_j \in \mathcal{M}_j^n$ quand la variable j est catégorielle, où $\mathcal{M}_j$ est l'ensemble des catégories de cette $j^{\text{ème}}$ variable.

L'approche ClusOfVar de classification de variables repose sur un critère d'homogénéité qui est basé sur les corrélations linéaires (au carré) et/ou le rapport de corrélation permettant de mesurer la "force" du lien existant entre les variables. Un algorithme de classification hiérarchique est fourni. Afin d'assister l'utilisateur dans son choix d'un nombre pertinent de classes à retenir pour la suite de l'étude, une approche de type "bootstrap" (qui évalue la stabilité des partitions) a été implémentée. De plus, un algorithme de partitionnement de type k-means est également disponible pour faire de la classification de variables.

2.1. *La critère d'homogénéité*

Nous donnons ici la définition de la variable synthétique permettant de résumer une classe (*cluster*). Puis, l'homogénéité d'une classe ainsi que l'homogénéité d'une partition en K classes seront présentées.

2.1.1. Définition de la variable synthétique de la classe C_k

Chaque classe C_k peut être résumée par une variable synthétique **numérique**, notée $\mathbf{y}_k$ dans la suite. Cette variable synthétique est définie ainsi :

$$\mathbf{y}_k = \arg\max_{\mathbf{u}\in\mathbb{R}^n} \left\{ \sum_{\substack{j \in C_k \\ j \text{ numérique}}} r^2(\mathbf{x}_j, \mathbf{u}) + \sum_{\substack{j \in C_k \\ j \text{ catégorielle}}} \eta^2(\mathbf{x}_j, \mathbf{u}) \right\},$$

où

- $r^2(\mathbf{x}_j, \mathbf{u}) \in [0, 1]$ est le carré de la corrélation linéaire (de type Pearson) entre la variable numérique $\mathbf{x}_j$ et la variable synthétique $\mathbf{u}$ (qui est numérique).
  Lorsque $r^2(\mathbf{x}_j, \mathbf{u})$ est proche de 1, la variable $\mathbf{x}_j$ est fortement liée linéairement à la variable synthétique $\mathbf{u}$, quel que soit le sens de la liaison (croissante ou décroissante).
- $\eta^2(\mathbf{x}_j, \mathbf{u}) \in [0, 1]$ est le rapport de corrélation entre la variable catégorielle $\mathbf{x}_j$ et la variable synthétique $\mathbf{u}$.
  Ce cœfficient mesure la part de la variance de $\mathbf{u}$ expliquée par les catégories de $\mathbf{x}_j$ et est défini par :

$$\eta^2(\mathbf{x}_j, \mathbf{u}) = \frac{\sum_{s\in\mathcal{M}_j} n_s(\bar{\mathbf{u}}_s - \bar{\mathbf{u}})^2}{\sum_{i=1}^{n}(u_i - \bar{\mathbf{u}})^2},$$

  où n_s est l'effectif de la catégorie s, $\bar{\mathbf{u}}_s$ la moyenne empirique de la variable $\mathbf{u}$ calculée sur les individus prenant à la catégorie s et $\bar{\mathbf{u}}$ est la moyenne empirique globale de $\mathbf{u}$.

Remarque. On peut montrer que la variable synthétique $\mathbf{y}_k$ de la classe C_k est la première composante principale de la méthode PCAmix appliquée aux données de la classe C_k, voir Chavent et al. [2012a]. La méthode PCAmix est brièvement présentée à la section suivante.

2.1.2. Homogénéité H de la classe C_k

L'homogénéité H de la classe C_k de variables est une mesure d'adéquation entre les variables de la classe C_k et sa variable synthétique (numérique) $\mathbf{y}_k$. Elle est définie de la manière suivante :

$$H(C_k) = \sum_{\substack{j \in C_k \\ j \text{ numérique}}} r^2(\mathbf{x}_j, \mathbf{y}_k) + \sum_{\substack{j \in C_k \\ j \text{ catégorielle}}} \eta^2(\mathbf{x}_j, \mathbf{y}_k) \qquad [1]$$

où $\mathbf{y}_k$ est la première composante principale de PCAmix appliquée aux variables appartenant à la classe C_k. On a alors :

$$H(C_k) = \lambda_1^k, \qquad [2]$$

où λ_1^k est la première valeur propre de PCAmix appliquée aux variables appartenant à la classe C_k.

Quelques commentaires.

- Le premier terme de [1] mesure le lien entre les variables numériques de la classe C_k et sa variable synthétique $\mathbf{y}_k$ indépendamment du signe de la liaison.
- Le second terme de [1] mesure le lien entre les variables catégorielles de la classe C_k et sa variable synthétique $\mathbf{y}_k$.
- Ainsi, l'homogénéité $H(C_k)$ de la classe C_k est **maximale** quand
  • toutes les variables numériques sont parfaitement corrélées (ou anti-corrélée) à $\mathbf{y}_k$ (les carrés des corrélations linéaires sont égales à 1),
  • et tous les rapports de corrélation des variables catégorielles sont également égales à 1 (correspondant à une liaison parfaite).
  Cela signifie que toutes les variables (numériques et catégorielles) de la classe C_k apportent la même information.

2.1.3. Homogénéité $\mathcal{H}$ de la partition $P_K = (C_1, \ldots, C_K)$

L'homogénéité $\mathcal{H}$ de la partition $P_K = (C_1, \ldots, C_K)$ est définie comme la somme des homogénéités de ses K classes :

$$\mathcal{H}(P_K) = \sum_{k=1}^{K} H(C_k). \qquad [3]$$

Quelques commentaires.

- L'homogénéité est **maximale** pour une partition de singletons (partition en p classes) avec $\mathcal{H}(P_p) = p$. En effet, nous avons $H(C_k) = 1$ pour chaque classe C_k quand C_k est un singleton (correspondant à une seule et unique variable qui est parfaitement liée à elle-même).
- Des relations [2] et [3], on déduit que $\mathcal{H}(P_K) = \sum_{k=1}^{K} \lambda_1^k$.

2.2. *Les algorithmes de classification de variables*

Le but d'un algorithme de classification est de trouver une partition d'un ensemble de p variables numériques et/ou catégorielles telle que les variables appartenant à une classe soient fortement reliées entre elles. En d'autres termes, l'objectif est de trouver une partition $P_K = (C_1, \ldots, C_K)$ qui maximise l'homogénéité $\mathcal{H}(P_K)$.

Deux algorithmes de classification de variables sont proposés dans le package **ClustOfVar** : un algorithme de classification hiérarchique et un algorithme de partitionnement (de type *k-means*). De plus, pour la classification hiérarchique, une procédure basée sur un re-échantillonnage de type *bootstrap* est proposée afin d'aider l'utilisateur à déterminer le nombre convenable de classes de variables à retenir pour l'analyse. Cette procédure repose sur la stabilité des partitions en $K = 2, 3, \ldots, p-1$ classes.

2.2.1. Classification hiérarchique des variables

2.2.1.1. Description de l'algorithme

Cet algorithme construit un ensemble de p partitions emboîtées de variables de la manière suivante.

1. On part d'une partition en p classes, c'est à dire avec une seule variable dans chaque classe.
2. Successivement , on agrège les deux classes A et B qui ont la plus petite dissimilarité d définie comme suit :
$$d(A, B) = H(A) + H(B) - H(A \cup B) = \lambda_1^A + \lambda_1^B - \lambda_1^{A \cup B}.$$
3. On s'arrête lorsque que la partition en une classe est obtenue, cette partition contenant les p variables.

Quelques commentaires.

– La dissimilarité d mesure la perte d'homogénéité observée quand les classes A et B sont fusionnées.
– En utilisant cette mesure, la nouvelle parition en $(p - l)$ classes maximise l'homogénéité $\mathcal{H}$ parmi toutes les partitions possibles en $(p - l)$ classes obtenues par agrégation de deux classes d'une partition en $(p - l + 1)$ classes.
– Dans le dendrogramme associé, la hauteur de la classe nouvelle $C = A \cup B$ est définie par $h(C) = d(A, B) \geq 0$.
– Notons que des inversions dans le dendrogramme peuvent être très rarement observées.

2.2.1.2. Les fonctions du package R **ClustOfVar** permettant de faire de la classification hiérarchique des variables

– La fonction `hclustvar` construit la hiérarchie.
– La fonction `plot` permet d'obtenir le dendrogramme de cette hiérarchie.
– La fonction `cutreevar` coupe la hiérarchie et extrait une partition en K classes où le nombre K de classes est donné par l'utilisateur.
– La fonction `rect.hclust` dessine des rectangles autour des branches du dendrogramme permettant de mettre en évidence les K classes obtenues.

2.2.1.3. Stabilité des partitions de variables

Dans le package **ClustOfVar**, une procédure évalue la stabilité des p partitions emboîtées du dendrogramme (obtenu avec la fonction `hclustvar`). Le graphique du critère de stabilité en fonction du nombre de classes peut aider l'utilisateur dans le choix (parfois complexe) d'un nombre approprié de classes à retenir pour la suite de son étude. Un "bon" choix pour ce nombre K de classes revient à retenir une partition en K classes qui est stable (au sens du critère de stabilité défini ci-après).

Le calcul de la stabilité d'une partition est fait dans la manière suivante.

1. B échantillons "bootstrap" des n observations sont générés et les B dendrogrammes associés sont obtenus avec la fonction `hclustvar`.

2. Les partitions de ces B dendrogrammes sont comparées avec les partitions de la hiérarchie initiale (c'est à dire calculée sur les données d'origine) en utilisant l'indice de Rand ajusté.
3. La stabilité de la partition est évaluée par la moyenne des valeurs des B indices de Rand ajusté.

Quelques commentaires.

– La stabilité d'une partition est implémentée dans la fonction `stability` du package **ClustOfVar**. Pour obtenir le graphique du critère de stabilité en fonction du nombre de classe, il suffit d'utiliser la fonction `plot`.
– L'indice de Rand et l'indice de Rand ajusté sont implémentés dans la fonction `rand` du package **ClustOfVar** (voir Hubert et Arabie [1985] pour plus de détails sur ces indices). Plus la valeur de l'indice est proche de 1, plus les partitions sont semblables.
– Le choix du nombre B d'échantillons "bootstrap" dépend du nombre p de variables et du nombre n d'observations des données d'origine afin d'avoir un temps de calcul du critère raisonnable. Plus p et n sont grands, plus l'utilisateur devra prendre une valeur raisonnable pour B (de l'ordre de 50 par exemple).

2.2.2. Classification de type k-means

Les algorithmes de partitionnement nécessitent la définition d'une **mesure de similarité** entre deux variables quel que soit son type (c'est à dire numérique ou catégorielle). A cette fin, la corrélation canonique (au carré) entre deux matrices de données $\mathbf{E}$ et $\mathbf{F}$ de dimensions respectives $n \times r_1$ et $n \times r_2$ peut être utilisée. Rappelons tout d'abord sa définition.

2.2.2.1. Définition de la corrélation canonique (au carré) ρ

Cette corrélation peut facilement être calculée de la manière suivante :

– si $\min(n, r_1, r_2) = n$,
alors $\rho(\mathbf{E}, \mathbf{F}) =$ première valeur propre de la matrice $\mathbf{EF}'\mathbf{FE}'$ de dimension $n \times n$;
– si $\min(n, r_1, r_2) = r_1$,
alors $\rho(\mathbf{E}, \mathbf{F}) =$ première valeur propre de la matrice $\mathbf{E}'\mathbf{FF}'\mathbf{E}$ de dimension $r_1 \times r_1$;
– si $\min(n, r_1, r_2) = r_2$,
alors $\rho(\mathbf{E}, \mathbf{F}) =$ première valeur propre de la matrice $\mathbf{F}'\mathbf{EE}'\mathbf{F}$ de dimension $r_2 \times r_2$.

Quelques commentaire sur ρ.

– Pour deux variables numériques $\mathbf{x}_i$ et $\mathbf{x}_j$, posons $\mathbf{E} = \tilde{\mathbf{x}}_i$ et $\mathbf{F} = \tilde{\mathbf{x}}_j$ où $\tilde{\mathbf{x}}_i$ et $\tilde{\mathbf{x}}_j$ sont les versions standardisées de $\mathbf{x}_i$ et $\mathbf{x}_j$. Dans ce cas, la corrélation canonique au carré est le carré du coefficient de corrélation linéaire (de Pearson) :

$$\rho(\mathbf{E}, \mathbf{F}) = r^2(\mathbf{x}_i, \mathbf{x}_j).$$

– Pour une variable catégorielle $\mathbf{x}_i$ et une variable numérique $\mathbf{x}_j$, posons $\mathbf{E} = \tilde{\mathbf{X}}_i$ et $\mathbf{F} = \tilde{\mathbf{x}}_j$ où $\tilde{\mathbf{X}}_i$ est la version standardisée de la matrice $\mathbf{G}_i$ des indicatrices des modalités de la variable catégorielle $\mathbf{x}_i$. Dans ce cas, la corrélation canonique au carré est la rapport de corrélation :

$$\rho(\mathbf{E}, \mathbf{F}) = \eta^2(\mathbf{x}_j, \mathbf{x}_i).$$

– Pour deux variables catégorielles $\mathbf{x}_i$ et $\mathbf{x}_j$, posons $\mathbf{E} = \tilde{\mathbf{X}}_i$ et $\mathbf{F} = \tilde{\mathbf{X}}_j$. Dans ce cas, la corrélation canonique au carré $\rho(\mathbf{E}, \mathbf{F})$ ne correspond à aucune mesure d'association connue. Son interprétation

est purement géométrique. Plus $\rho(\mathbf{E}, \mathbf{F})$ est proche de 1, plus les deux sous-espaces linéaires de $\mathbb{R}^n$ engendrés par les matrices $\mathbf{E}$ et $\mathbf{F}$ sont proches, et ainsi les deux variables catégorielles $\mathbf{x}_i$ et $\mathbf{x}_j$ apportent une information similaire.

2.2.2.2. L'algorithme de type k-means de classification des variables

Les algorithmes de classification de type k-means construisent une partition en K classes (avec K spécifié a priori par l'utilisateur) de la manière suivante.

– ETAPE D'INITIALISATION. Au choix :
  ↪ Une partition en K classes est fournie en entrée par l'utilisateur. Par exemple, la partition initiale peut être obtenue en coupant le dendrogramme de la hiérarchie en K classes.
  ↪ Une partition aléatoire est obtenue comme suit :
    1. Sélection aléatoire de K variables comme centres initiaux.
    2. Allocation de chaque variable à la classe ayant le centre initial le plus proche : la similarité entre la variable et un centre initial est calculée en utilisant la corrélation canonique ρ.

– ETAPES DE REPRÉSENTATION ET D'ALLOCATION.
  Répéter :
    1. **Une étape représentation :** construire la variable synthétique $\mathbf{y}_k$ de chaque classe C_k en appliquant PCAmix aux variables de la classe C_k.
    2. **Une étape d'allocation :** affecter chaque variable à la classe la plus proche où la classe la plus proche est celle dont la variable synthétique est la plus proche au sens de la corrélation canonique ρ.

  Arrêter lorsqu'il n'y a plus de changements dans la partition ou bien lorsqu'un nombre maximal d'itérations (fixé par l'utilisateur) est atteint.

Quelques commentaires.

– Cet algorithme de type k-means est implémenté dans la fonction `kmeansvar` dans le package **ClustOfVar**.
– Cette procédure itérative fournit une partition P_K en K classes qui maximise $\mathcal{H}$, cependant cet optimum n'est souvent qu'un optimum local et peut dépendre de la partition initiale (étape d'initialisation).
  Une solution pour éviter ce problème et pour réduire l'influence du choix arbitraire de la partition initiale est de considérer plusieurs initialisations aléatoires. Dans ce cas, les étapes d'initialisation aléatoire, de représentation et d'allocation sont répétées. L'idée est alors de retenir comme partition finale celle qui a permis d'obtenir la plus grande valeur de $\mathcal{H}$, voir l'argument `nstart` de la fonction `kmeansvar`.

3. Analyse en composantes principales pour données mixtes : l'approche PCAmix

En analyse des données multidimensionnelles (ou multivariées), il existent diverses méthodes standards. On peut mentionner en autres les méthodes suivantes :

– l'analyse en composantes principales (PCA pour *Principal Component Analysis* en anglais) pour des données numériques,
– l'analyse factorielle multiple des correspondances (MCA pour *Multiple Correspondence Analysis* en anglais) pour des données catégorielles,
– l'analyse factorielle multiple (MFA pour *Multiple Factor Analysis* en anglais) ou encore la méthode STATIS pour les données multi-tableaux (encore appelées multi-voies), les données doivent cependant être de même nature (numérique or catégorielle) au sein d'un même tableau.

Nous donnons ici une brève description de la méthode PCAmix qui est implémentée dans le package R **PCAmixdata**. Cette méthode est utile pour traiter des jeux de données contenant des variables numériques et/ou catégorielles. PCAmix permet ainsi de faire l'analyse en composantes principales pour des données mixtes.

Introduisons quelques notations.
– Soit n le nombre d'individus (observations).
– Soit $\mathbf{X}_1$ la matrice de dimension $n \times p_1$ contenant des données associées à p_1 variables **numériques** mesurées sur les n individus.
– Soit $\mathbf{X}_2$ une matrice de dimension $n \times p_2$ contenant des données associées à p_2 variables **catégorielles** mesurées sur les n individus. Notons par m le nombre total de catégories (appelées aussi modalités ou niveaux) de ces p_2 variables catégorielles.
– Enfin, rappelons que $p = p_1 + p_2$ est le nombre total de variables considérées.

3.1. *Présentation de PCAmix*

Les objectifs principaux de PCAmix sont de fournir
– un ensemble de nouvelles variables **numériques** orthogonales (variables synthétiques) appelées composantes principales ;
– des sorties graphiques, permettant de mettre en évidence (visualiser) des similarités entre les observations et entre les variables numériques et/ou les modalités des variables catégorielles (par le biais projections sur les plans factoriels associés).

L'approche PCAmix est fondée sur une décomposition aux valeurs singulières généralisée (GSVD, pour *Generalized Singular Value Decomposition* en anglais). L'algorithme correspondant est fourni ci-après.

Les idées sous-jacentes derrière la construction des composantes principales sont les suivantes. Les composantes principales (notées $\mathbf{f}_\alpha \in \mathbb{R}^n$) sont des combinaisons linéaires *non corrélées* (i.e. orthogonales) des colonnes de la matrice $\mathbf{Z}$ (version "standardisée" des matrices de données d'origine $\mathbf{X}_1$ et $\mathbf{X}_2$, voir l'algorithme pour plus de détails sur ce pré-traitement des données) avec
– une dispersion maximale au sens de la variance :

$$\text{Var}(\mathbf{f}_\alpha) = \lambda_\alpha,$$

– une liaison maximale avec les variables d'origine :

$$\sum_{j=1}^{p_1} r^2(\mathbf{x}_j, \mathbf{f}_\alpha) + \sum_{j=p_1+1}^{p_2} \eta^2(\mathbf{x}_j, \mathbf{f}_\alpha) = \lambda_\alpha,$$

où r^2 (resp. η^2) est respectivement la corrélation linéaire au carré (resp. le rapport de corrélation), et λ_α est la valeur propre associée à la composante principale $\mathbf{f}_\alpha$ (voir l'algorithme pour plus de détails).

Remarque. La méthode PCAmix inclut comme cas particuliers
- l'analyse en composantes principales (PCA) lorsque les données ne concernent que des variables numériques,
- l'analyse factorielle multiple des correspondances (MCA) lorsque les données concernent exclusivement des variables catégorielles.

Quelques commentaires.
- Chaque valeur propre λ_α est la variance de la composante principale $\mathbf{f}_\alpha$, qui est la $\alpha^{\text{ème}}$ colonne de la matrice $\mathbf{F}$ donnée dans l'algorithme.
- Les composantes principales peuvent être vues comme les scores des individus.
- Il est également possible d'associer des scores aux variables (donnés dans la matrice $\mathbf{A}$, voir l'algorithme pour plus de détails).
  Chaque score $a_{j\alpha}$ d'une variable numérique $\mathbf{x}_j$ est sa corrélation linéaire avec la $\alpha^{\text{ème}}$ composante principale $\mathbf{f}_\alpha$. Chaque score $a_{s\alpha}$ d'une catégorie s d'une variable catégorielle est la moyenne des scores des observations de cette catégorie.
- La contribution $c_{j\alpha}$ de la $j^{\text{ème}}$ variable à la construction de la $\alpha^{\text{ème}}$ composante principale component est définie par le carré de la corrélation linéaire entre cette variable $\mathbf{x}_j$ et la composante $\mathbf{f}_\alpha$ si $\mathbf{x}_j$ est numérique, et par le rapport de corrélation entre cette variable $\mathbf{x}_j$ et la composante $\mathbf{f}_\alpha$ si $\mathbf{x}_j$ est catégorielle. Plus précisemment,

$$\begin{cases} c_{j\alpha} = a_{j\alpha}^2 & \text{si la variable } \mathbf{x}_j \text{ est numérique,} \\ c_{j\alpha} = \sum_{s\in\mathcal{M}_j} \frac{n}{n_s} a_{s\alpha}^2 & \text{si la variable } \mathbf{x}_j \text{ est catégorielle.} \end{cases}$$

  Les contributions sont également appelées *squared loadings* en anglais (et dans les sorties numériques et graphiques associées).

Ces diverses quantités sont utilisées afin de produire les divers graphiques de la méthode PCAmix permettant de visualiser les similarités entre les observations et entre les variables numériques et/ou les modalités. Ces graphiques sont des projections sur les plans factoriels associés : *individual component maps, correlation circles, levels component map, squared loadings map*, voir la section "Application" pour des exemples de graphiques et leurs interprétations.

3.2. *L'algorithme de PCAmix*

Cet algorithme fonctionne en trois étapes principales.

1. ETAPE DE PRÉ-TRAITEMENT DES DONNÉES.
   (a) Construire la matrice (numérique) des données $\mathbf{Z} = (\mathbf{Z}_1|\mathbf{Z}_2)$ de dimension $n \times (p_1 + m)$ où $\mathbf{Z}_1$ est la version standardisée de $\mathbf{X}_1$ et $\mathbf{Z}_2$ est matrice centrée des indicatrices des modalités issues de $\mathbf{X}_2$.
   (b) Construire la matrice diagonale $\mathbf{N}$ des poids des lignes (observations). Les n lignes sont pondérées par $\frac{1}{n}$, c'est à dire $\mathbf{N} = \text{diag}(\frac{1}{n},\ i = 1, \ldots, n)$.
   (c) Construire la matrice diagonale $\mathbf{M}$ des poids des colonnes (variables numériques ou indicatrices des modalités des variables catégorielles).
   $\hookrightarrow$ Les p_1 premières colonnes (variables numériques) sont pondérées par 1.
   $\hookrightarrow$ Les m dernières colonnes (modalités) sont pondérées par $\frac{n}{n_s}$, avec n_s le nombre d'observations de la modalité s.

Remarque. L'inertie totale de $\mathbf{Z}$ est égale à $p_1 + m - p_2$.

2. ETAPE DE GSVD.
   La GSVD de $\mathbf{Z}$ avec les métriques diagonales des poids $\mathbf{N}$ et $\mathbf{M}$ fournit la décomposition suivante :
   $$\mathbf{Z} = \mathbf{U}\Lambda\mathbf{V}^t$$
   où
   - $\Lambda = \text{diag}(\sqrt{\lambda_1}, \ldots, \sqrt{\lambda_r})$ est la matrice diagonale de dimension $r \times r$ des valeurs singulières de $\mathbf{ZMZ}^t\mathbf{N}$ et $\mathbf{Z}^t\mathbf{NZM}$, avec r désignant le rang $\mathbf{Z}$;
   - $\mathbf{U}$ est la matrice de dimension $n \times r$ des r premiers vecteurs propres de $\mathbf{ZMZ}^t\mathbf{N}$ tels que $\mathbf{U}^t\mathbf{NU} = \mathbb{I}_r$;
   - $\mathbf{V}$ est la matrice de dimension $p \times r$ des r premiers vecteurs propres de $\mathbf{Z}^t\mathbf{NZM}$ tels que $\mathbf{V}^t\mathbf{MV} = \mathbb{I}_r$.
3. ETAPE DE CALCULS DES SCORES.
   (a) L'ensemble des scores (factoriels) pour les lignes (individus) est donné par :
   $$\mathbf{F} = \mathbf{U}\Lambda.$$
   (b) L'ensemble des scores (factoriels) pour les colonnes (variables) est donné par :
   $$\mathbf{A} = \mathbf{MV}\Lambda = \begin{pmatrix} \mathbf{A}_1 \\ \mathbf{A}_2 \end{pmatrix} \begin{matrix} \} p_1 \\ \} m \end{matrix}$$
   où $\mathbf{A}_1$ (resp. $\mathbf{A}_2$) contient les scores des p_1 variables numériques (resp. les scores des m catégories des p_2 variables catégorielles).

Remarque. Ce résultat est légèrement différent de celui de la PCA standard où $\mathbf{A} = \mathbf{V}\Lambda$.

3.3. *Prédiction des composantes principales*

Chaque composante principale $\mathbf{f}_\alpha$ s'écrit comme une combinaison linéaire des colonnes de $\mathbf{X} = (\mathbf{X}_1|\mathbf{G})$ où $\mathbf{X}_1$ est la matrice des données numériques et $\mathbf{G}$ est la matrice des indicatrices des modalités des variables catégorielles issues de $\mathbf{X}_2$:

$$\mathbf{f}_\alpha = \beta_0 + \sum_{j=1}^{p_1+m} \beta_j \mathbf{x}_j \quad \text{avec} \quad \begin{cases} \beta_0 = -\sum_{k=1}^{p_1} v_{k\alpha} \frac{\bar{\mathbf{x}}_k}{s_k} - \sum_{k=p_1+1}^{p_1+m} v_{k\alpha}, \\ \beta_j = v_{j\alpha} \frac{1}{s_j} \quad \text{pour} \quad j = 1, \ldots, p_1, \\ \beta_j = v_{j\alpha} \frac{n}{n_j} \quad \text{pour} \quad j = p_1 + 1, \ldots, p_1 + m. \end{cases}$$

Ces cœfficients peuvent alors être utilisés par exemple pour représenter une nouvelle observation sur les plans factoriels.

4. Illustration sur les données réelles BCI : PCAmix et classification des variables

4.1. *Méthodologie*

L'objectif est ici de montrer le fonctionnement des méthodes de classification de variables et d'analyse en composantes principales sur le jeu de données mixtes issues de l'expérimentation BCI décrite à la section 1.

Ces deux approches, PCAmix et ClustOfVar, vont permettre de mettre en évidence les liens existant entre les différentes variables et de résumer l'information par le biais de la constructions de nouvelles variables synthétiques numériques (qui peuvent ensuite être utilisées pour poursuivre une étude statistique, comme par exemple faire une classification des individus). Rappelons que les $K < p$ nouvelles variables synthétiques obtenues sont

– les K premières composantes principales pour la méthode PCAmix ;
– les représentants des K classes retenues pour l'approche ClustOfVar.

Notons également que :

– les variables synthétiques obtenues avec PCAmix sont des combinaisons linéaires de l'ensemble des p variables disponibles et sont orthogonales entre elles ;
– les variable synthétiques obtenues avec ClustOfVar sont des combinaisons linéaires des seules variables de la classe correspondante. Cela rend ainsi leur interprétation plus aisée, mais ces K variables synthétiques ne sont pas orthogonales entre elles.

Nous fournissons ci-après les commandes R (basées sur l'usage des packages **PCAmixdata** et **ClustOfVar**) et nous commentons les sorties numériques et graphiques obtenues. Les deux packages R **PCAmixdata** et **ClustOfVar** sont disponibles sur le CRAN[1]. Une nouvelle version de ces packages ainsi que des vignettes (en anglais) décrivant les fonctions et leur utilisation sont également sur GitHub :

```
https ://github.com/chavent/PCAmixdata
https ://github.com/chavent/ClustOfVar
```

Débutons par "charger" les deux packages dans R :

```
> library(PCAmixdata)
> library(ClustOfVar)
```

Dans les codes R ci-après, les variables numériques (resp. catégorielles) sont stockées dans les *data frame* appelées `donneesQuanti` (resp. `donneesQuali`). La dimension de ces jeux de données est obtenue par la fonction `dim`, et un extrait de ces données peut être obtenu par le biais de la fonction `head`.

```
> dim(donneesQuanti)
[1] 72 17

> head(donneesQuanti)
  Pct.Erreur Pct.errZONE.ROUGE Pct.errZONE.VERT Pct.errZONE.JAUNE max.Pct.cal ...
1      41.67             33.33               25               100         100
2       0.00              0.00                0                 0         100
3      16.67             16.67                0                50         100
4       0.00              0.00                0                 0         100
5       0.00              0.00                0                 0         100
6       8.33             16.67                0                 0         100
....
  Frustration WKL.total PERC1.Symboles PERC2.Distance PERC3.lig.col
1        5.33     63.33              6              5             2
2        0.67     39.33              0              0             2
3        1.33     48.00              1              1             2
4        0.00     18.00              2              3             3
5        0.00     18.67              2              3             3
6        4.00     26.00              2              2             4
```

1.https ://cran.r-project.org/

```
> dim(donneesQuali)
[1] 72  8
> head(donneesQuali)
  Condition Taille niv.complex niv.confort niv.stress niv.ctrl niv.fatig niv.pref
1    Mobile      G       Cplx3       Conf1      Stre3    Ctrl2     Fati3    Pref1
2    Mobile      I       Cplx1       Conf2      Stre2    Ctrl3     Fati2    Pref3
3    Mobile      P       Cplx2       Conf3      Stre1    Ctrl1     Fati1    Pref2
4    Mobile      G       Cplx2       Conf2      Stre3    Ctrl1     Fati3    Pref2
5    Mobile      I       Cplx1       Conf3      Stre2    Ctrl3     Fati2    Pref3
6    Mobile      P       Cplx3       Conf1      Stre1    Ctrl2     Fati1    Pref1
```

On retrouve bien $p_1 = 17$ variables numériques et $p_2 = 8$ variables catégorielles mesurées sur $n = 72$ expériences BCI.

4.2. *Application de la méthode PCAmix*

Pour appliquer la méthode PCAmix à nos données, il suffit d'utiliser la fonction `PCAmix` comme indiqué ci-après, les résultats étant stockés dans l'objet `resPCA`.

```
> resPCA <- PCAmix(X.quanti = donneesQuanti, X.quali = donneesQuali,  graph = FALSE)
```

Les sorties numériques ci-dessous nous permettent de voir quelle est la part de l'information apportée par chaque axe factoriel.

```
> round(resPCA$eig,digit=2)
       Eigenvalue Proportion Cumulative
dim 1        8.63      26.97      26.97
dim 2        3.86      12.05      39.02
dim 3        2.72       8.51      47.53
dim 4        1.97       6.14      53.68
dim 5        1.80       5.63      59.31
dim 6        1.63       5.11      64.41
dim 7        1.39       4.34      68.76
dim 8        1.23       3.86      72.61
dim 9        0.99       3.09      75.71
...
```

L'étude des premières valeurs propres (voir `res.PCA$eig` tronqués aux 9 premières dimensions) montrent que les deux premières composantes principales expliquent près de 40% de la variance totale, chacune des composantes suivantes n'apporte marginalement que peu d'information complémentaire (par exemple 8,5% par l'axe factoriel 3, et 6,1% pour l'axe factoriel 4). Dans un soucis de clarté et de concision, les commentaires suivants se focaliseront uniquement sur les premières composantes principales de PCAmix.

Les lignes de commandes ci-après permettent de générer respectivement les différents graphiques suivants : *individual component maps, correlation circles, levels component map, squared loadings map*

- le cercle des corrélations (*correlation circles*, voir la figure **3**) qui correspond à la projection des variables quantitatives sur le plan factoriel 1-2 ; seules les variables bien représentées sur le plan sont représentées (i.e. celle ayant une qualité de représentation supérieure à 50%) ;
- la projection des modalités des variables catégorielles sur le plan factoriel 1-2 (*levels component map*, voir la figure **4**) ;

– la projection des individus sur le plan factoriel 1-2 (*individual component maps*, voir la figure **5**) ; les individus sont colorés selon la taille de l'écran (graphique du haut) ou selon la condition (graphique du bas).

```
> plot(resPCA,axes=c(1,2),choice="cor",lim.cos2.plot=0.5)
> plot(resPCA,axes=c(1,2),choice="levels")
> plot(resPCA,axes=c(1,2),choice="ind",coloring.ind=Taille,posleg = "bottomleft")
> plot(resPCA,axes=c(1,2),choice="ind",coloring.ind=Condition,posleg = "bottomleft")
```

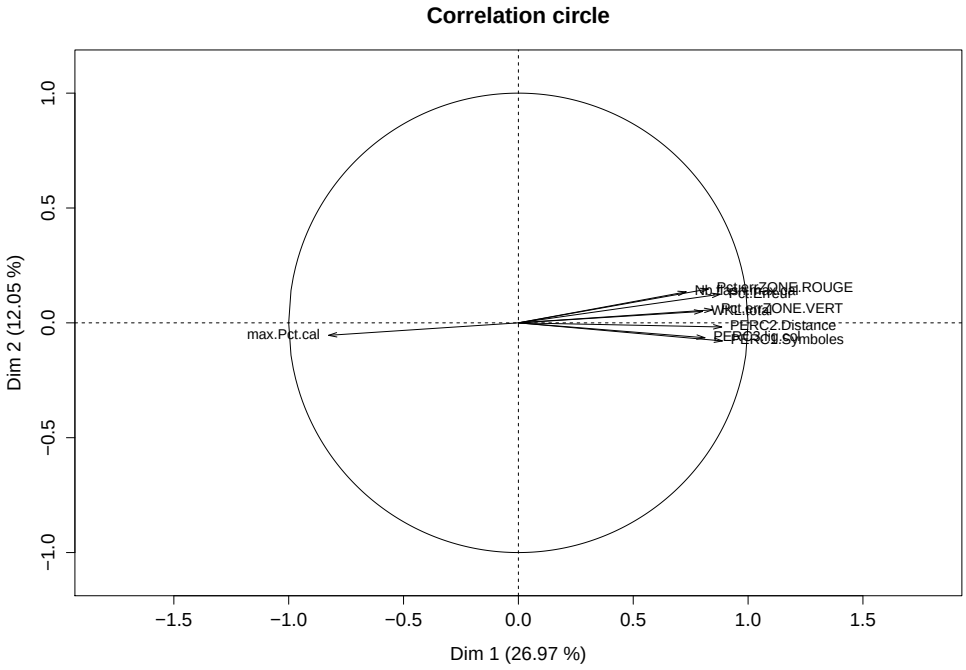

Figure 3. Projection des variables quantitatives sur le plan factoriel 1-2.

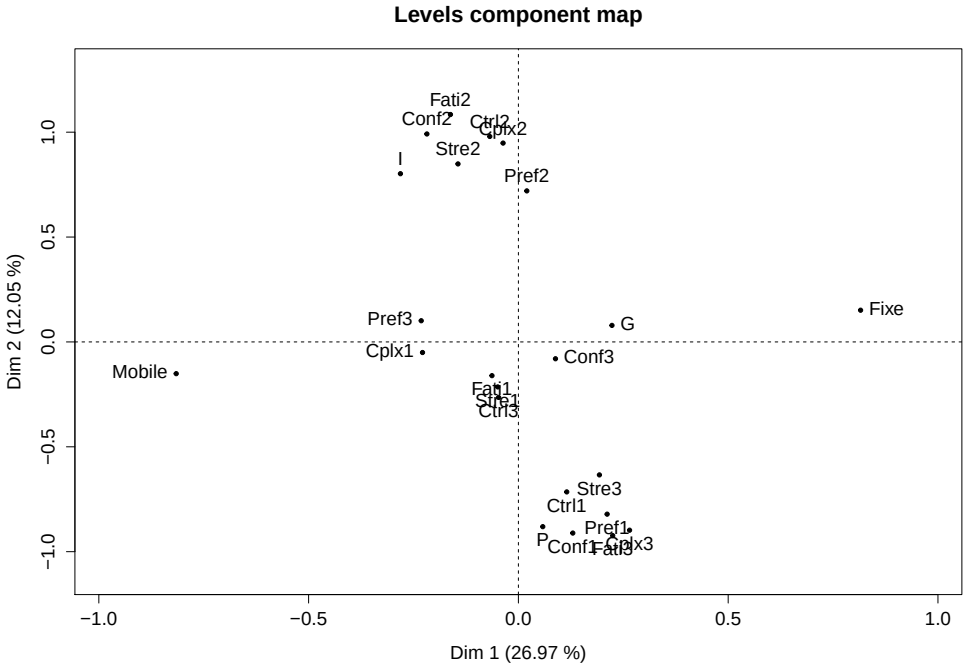

Figure 4. Projection des modalités des variables catégorielles sur le plan factoriel 1-2.

Les sorties numériques ci-dessous (*squared loading*) fournissent les qualités de représentation de chacune des variables (numériques ou catégorielles) sur les cinq premiers axes factoriels.

```
> round(resPCA$sqload,digit=3)
                    dim 1 dim 2 dim 3 dim 4 dim 5
Pct.Erreur          0.772 0.016 0.017 0.116 0.008
Pct.errZONE.ROUGE   0.684 0.022 0.014 0.112 0.008
Pct.errZONE.VERT    0.715 0.003 0.005 0.109 0.007
Pct.errZONE.JAUNE   0.311 0.020 0.007 0.064 0.006
max.Pct.cal         0.683 0.003 0.006 0.039 0.001
Nb.flash.max.cal    0.534 0.018 0.037 0.001 0.004
FATIGUE.VAS         0.182 0.000 0.134 0.153 0.099
Exigence.mentale    0.268 0.005 0.014 0.115 0.139
Exigence.physique   0.007 0.003 0.094 0.124 0.299
Exigence.temporelle 0.134 0.008 0.006 0.046 0.013
Effort              0.251 0.004 0.001 0.160 0.007
Degree.Performance  0.188 0.028 0.038 0.054 0.074
Frustration         0.164 0.052 0.006 0.015 0.138
WKL.total           0.643 0.002 0.002 0.162 0.000
PERC1.Symboles      0.789 0.006 0.005 0.030 0.000
PERC2.Distance      0.784 0.000 0.009 0.009 0.011
PERC3.lig.col       0.660 0.004 0.003 0.056 0.002
Condition           0.665 0.023 0.015 0.003 0.001
Taille              0.044 0.475 0.002 0.152 0.283
niv.complex         0.041 0.569 0.502 0.069 0.111
niv.confort         0.024 0.607 0.249 0.004 0.027
niv.stress          0.020 0.389 0.351 0.132 0.029
niv.ctrl            0.007 0.514 0.587 0.068 0.039
niv.fatig           0.027 0.685 0.341 0.021 0.271
niv.pref            0.033 0.401 0.279 0.153 0.224
```

Pour la dimension 1 (axe horizontal sur les figures **3** et **4**), il apparait clairement que 10 variables ont une qualité de représentation très élevée (supérieure à 0.53 c'est à dire qu'au minima 53% de l'information d'une variable est expliquée par le premier axe factoriel). Les 10 variables les mieux représentées sont, par ordre d'importance : `PERC1.Symboles` (0.78) - `PERC2.Distances` (0.78) - `Pct.Erreur` (0.77) - `Pct.errZONE.VERT` (0.71) - `Pct.errZONE.ROUGE` (0.68) - `max.Pct.cal` (0.68) - `Condition` (0.66) - `PERC3.lig.col` (0.66) - `WKL.total` (0.64) - `Nb.flash.max.cal` (0.53). La figure **3** permet clairement de montrer que la dimension 1 peut être interprétée comme un axe de performance (toutes les variables très corrélées positivement avec cet axe étant celles mesurant les pourcentages d'erreurs et celle corrélée négativement étant celle mesurant la qualité de la calibration). Plus une observation (voir les graphiques de la figure **5**) est projetée à droite (resp. à gauche) sur plus cette observation correspond à une expérience durant laquelle les taux d'erreurs ont été élevés (resp. faible). Ainsi, la dimension 1 permet d'expliquer l'efficacité. Ce premier axe factoriel oppose également les deux modalités de la variable "Condition" (fixe vs. mobile), voir la figure **4**, la modalité "fixe" étant projetée à droite et la modalité "mobile" étant projetée à gauche. Ceci est aussi très clairement visible sur le graphique du bas de la figure **5**. Ainsi, en synthétisant les conclusions précédentes, on peut en déduire que les expériences faites dans la condition "fixe" sont associées à une performance faible (forts pourcentages d'erreurs) contrairement aux expériences réalisées dans la condition "mobile".

Pour la dimension 2 (axe vertical sur les figures **3** et **4**), il apparait que seules 5 variables ont une qualité de représentation élevée (supérieure à 0.47 c'est à dire qu'au minima 47% de l'information d'un variable est expliquée par la dimension 1). Les 5 variables les mieux représentées sont, par ordre d'importance : `niv.fatig` (0.68), `niv.confort` (0.60), `niv.complex` (0.56), `niv.ctrl` (0.51), `Taille` (0.47). Ainsi,

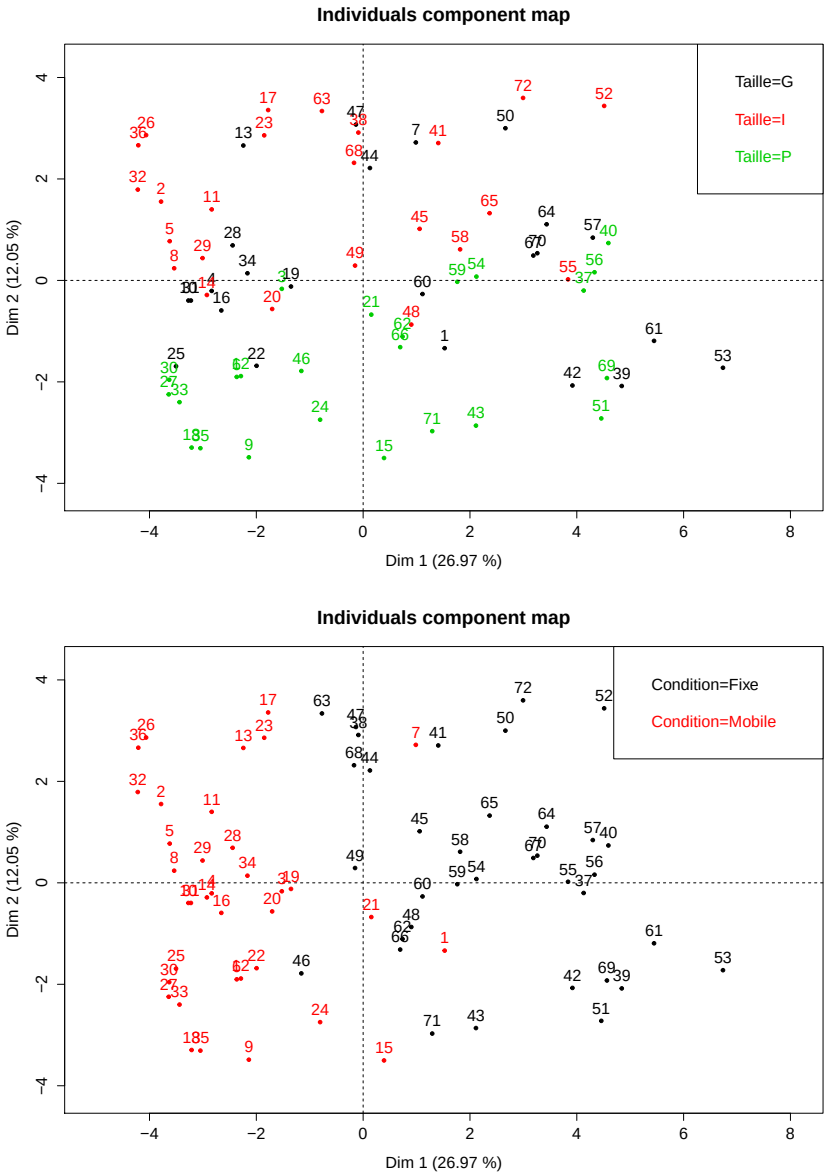

Figure 5. Projections des individus sur le plan factoriel 1-2. Les individus sont colorés selon la taille de l'écran (en haut) ou selon la condition (en bas).

la dimension 2 permet d'expliquer uniquement la satisfaction associée à la variable "Taille" (Petit vs Intermédiaire vs Grand). La Figure **4** permet clairement de montrer que la dimension 2, permet d'opposer les tailles de l'écran "Petit" et "Intermédiaire". En effet, la taille "Intermédiaire" est associée avec une satisfaction intermédiaire alors que la taille "Petit" est associée avec une satisfaction faible. Ceci est parfaitement confirmé en regardant la figure **5** (graphique du haut) dans laquelle on voit bien que les expériences faites avec un petit écran sont projetées en bas de l'axe 2, contrairement à celles faites avec un écran intermédiaire qui sont projetées en haut de l'axe 2.

Il est à noter que certaines variables sont peu ou pas expliquées par les dimensions 1 et 2 : `Pct.errZONE.JAUNE`, `FATIGUE.VAS`, `Exigence.mentale`, `Exigence.physique`, `Exigence.temporelle`, `Effort`, `Degree.Performance`, `Frustration`, `niv.stress` et `niv.pref`. Les informations apportées par ces variables sont disponibles sur les axes factoriels d'ordre supérieur, cela signifie implicitement que ce n'est pas celles qui permettent de différencier les plus observations (expériences BCI) traitées.

4.3. *Application des approches de classification de variables*

Nous allons dans un premier temps faire une classification hiérarchique des variables, puis nous ferons ensuite une classification de type k-means que nous comparerons à la première partition obtenue. On rappelle ici que l'objectif est de réunir dans une même classe des variables fortement liées entre elles (i.e. apportant le même type d'information).

4.3.1. Utilisation de la classification hiérarchique

La classification hiérarchique des données BCI considérées est obtenue par le biais de la fonction `hclustvar` du package R **ClustOfVar**. Les résultats associés sont stockés dans l'objet `tree`.

```
> tree <- hclustvar(X.quanti = donneesQuanti, X.quali = donneesQuali)
```

Afin de déterminer un nombre K raisonnable de classes à considérer pour la suite de l'étude, nous pouvons utiliser le crière de stabilité proposé précédemment. La fonction `plot` permet d'avoir une représentation graphique de ce critère (voir la figure 6).

```
> stab <- stability(tree,B=100)
> plot(stab)
```

Au vu du graphique de la figure 6, il apparaît raisonnable de retenir $K = 3$ pour l'interprétation, même si une partition en 2 classes est naturellement plus stable, cette dernière reste trop grossière pour la suite de l'interprétation. Une partition en un nombre plus grand de classes n'est pas plus stable (sauf à prendre un très grand nombre de classes ($K \geq 14$) ce qui n'est pas non plus intéressant en terme de résumé d'information).

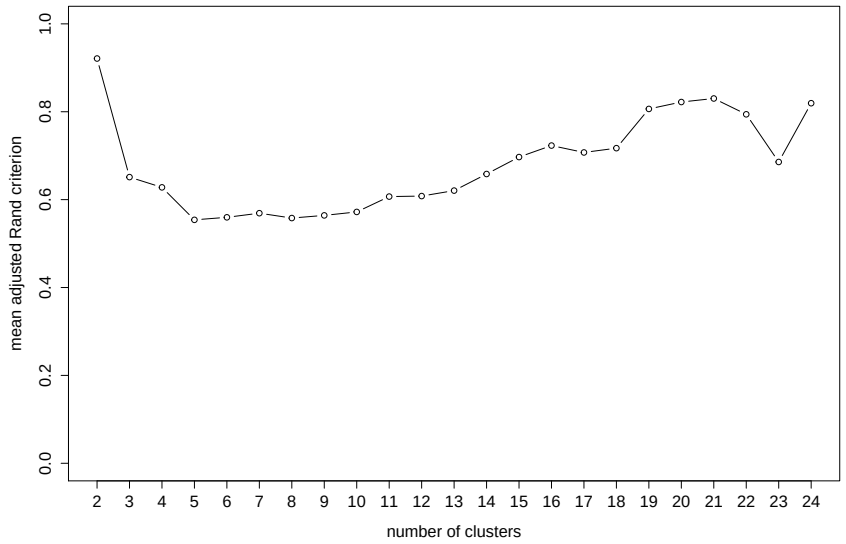

Figure 6. Critère de stabilité des partitions (de variables) obtenues par la classification hiérarchique.

Une fois déterminé le nombre de classes, il est possible de tracer le dendrogramme avec la fonction `plot` et de les visualiser via la fonction `rect.hclust`. Le graphique correspondant est visible à la figure 7. La partition en $K = 3$ classes ainsi obtenue est stockée dans l'objet `partition1` en utilisant la fonction `cutreevar` (qui permet de découper le dendrogramme en 3 classes).

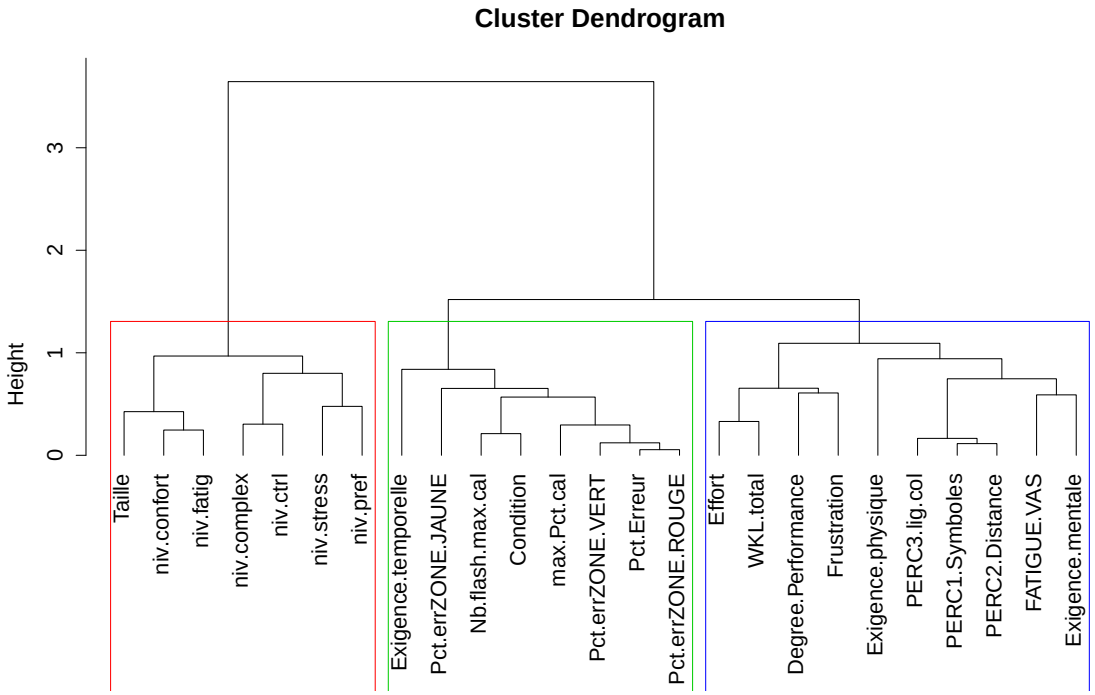

Figure 7. Dendrogramme obtenu par la la classification hiérarchique des variables. Les rectangles colorés correspondent aux trois groupes (clusters) retenus de variables.

```
> plot(tree)
> rect.hclust(tree,k=3,border=2:4)
> partition1 <- cutreevar(tree,k=3)
```

Une description des $K = 3$ classes de cette partition est disponible via la function `summary`. On peut ainsi voir les liaisons qui existent entre les variables de chaque classe et la variable synthétique $\mathbf{y}_k$ résumant la classe : plus le *squared loading* (qui est soit $r^2(\mathbf{x}_j, \mathbf{y}_k)$ pour une variable $\mathbf{x}_j$ numérique, soit $\eta^2(\mathbf{x}_j, \mathbf{y}_k)$ pour une variable $\mathbf{x}_j$ catégorielle) est proche de 1, plus la variable $\mathbf{x}_j$ est bien représentée dans cette classe. Pour les variables quantitatives (uniquement), on dispose également de la corrélation linéaire $r(\mathbf{x}_j, \mathbf{y}_k)$ qui nous informe si la variable $\mathbf{x}_j$ est corrélée positivement ou négativement avec la variable synthétique $\mathbf{y}_k$. Pour les variables catégorielles, cette information n'a pas de sens et la valeur `NA` (Not Available) est alors affichée. Dans chaque groupe, les variables sont classées selon leur qualité de représentation dans le groupe.

```
> summary(partition1)

Call:
cutreevar(obj = tree, k = 3)

Data:
   number of observations:  72
   number of  variables:  25
        number of numerical variables:  17
        number of categorical variables:  8
   number of clusters:  3

Cluster  1 :
                    squared loading correlation
Pct.Erreur                     0.92       -0.96
Pct.errZONE.ROUGE              0.83       -0.91
Pct.errZONE.VERT               0.83       -0.91
```

```
max.Pct.cal                    0.76        0.87
Condition                      0.73          NA
Nb.flash.max.cal               0.60       -0.78
Pct.errZONE.JAUNE              0.39       -0.63
Exigence.temporelle            0.19       -0.44

Cluster  2 :
                   squared loading correlation
WKL.total                    0.842       -0.92
PERC1.Symboles               0.809       -0.90
PERC2.Distance               0.735       -0.86
PERC3.lig.col                0.727       -0.85
Exigence.mentale             0.429       -0.65
Effort                       0.383       -0.62
FATIGUE.VAS                  0.337       -0.58
Degree.Performance           0.240       -0.49
Frustration                  0.215       -0.46
Exigence.physique            0.041       -0.20

Cluster  3 :
            squared loading correlation
niv.fatig              0.74          NA
niv.confort            0.64          NA
niv.complex            0.60          NA
niv.ctrl               0.54          NA
Taille                 0.48          NA
niv.pref               0.40          NA
niv.stress             0.39          NA

Gain in cohesion (in %):  31.55
```

Commentons maintenant les $K = 3$ groupes (ou clusters) de variables obtenus par la classification hiérarchique (figure 7) :

- le premier groupe de variables (cluster 1) est un regroupement d'efficacité puisque que l'on retrouve toutes les variables performances en relation avec la variable Condition. La condition Fixe est associée avec les performances les plus faibles et la condition Mobile est associée avec les performances les plus élevées.
- le second groupe de variables (cluster 2) est un regroupement d'efficience puisque que l'on retrouve exclusivement les variables efficiences.
- le troisième groupe de variables (cluster 3) est un regroupement de satisfaction en lien avec la variable Taille.

Il est possible de récupérer les représentants de chacun des $K = 3$ groupes de variables, à savoir $\mathbf{y}_1$, $\mathbf{y}_2$ et $\mathbf{y}_3$, qui sont les colonnes de l'objet ci-dessous :

```
> partition1$scores
    cluster1  cluster2   cluster3
1 -0.4362219 -1.294506  1.6421770
2  2.0026576  2.812696 -1.8653045
3  1.0494630  1.083590  0.2231275
4  2.2590878  2.110798 -0.2314623
5  2.5714801  2.211659 -1.2685148
6  2.0125668  1.622764  1.4999770
...
```

Ces variables synthétiques sont des scores permettant de résumer numériquement chaque groupe et donc de quantifier en une seule valeur par observation (i.e. expérience) l'information apportée par chaque

groupe de variables :

- $\mathbf{y}_1$ est un score d'efficacité dépendant de la condition expérimentale ("fixe" vs "mobile"),
- $\mathbf{y}_2$ est un score de satisfaction dépendant de la taille de l'écran,
- $\mathbf{y}_3$ est un score d'efficience ne dépendant pas a priori du protocole expérimental (condition et taille de l'écran).

4.3.2. Utilisation de la classification de type k-means

Il est aussi possible de faire une classification de variables de type k-means (via la fonction `kmeansvar`). Le paramètre `init` correspond au choix du nombre de classes désiré par l'utilisateur et le paramètre `nstart` est le nombre d'initialisations aléatoires utilisées dans la première étape de l'algorithme k-means. Les résultats de cette classification sont stockés dans l'objet `partition2`.

```
> partition2 <- kmeansvar(X.quanti = donneesQuanti, X.quali = donneesQuali,
                          init = 3, nstart = 200)
```

Il est utile de comparer les deux partitions de variables (en trois classes) obtenues avec la méthode de classification hiérarchique (`partition1`) et la méthode de type k-means (`partition2`). Pour cela, nous pouvons croiser les deux partitions avec la table de contingence ci-dessus (avec la fonction `table`) et calculer l'index de Rand :

```
> table(partition1$cluster,partition2$cluster)
     1  2  3
  1  8  0  0
  2  0 10  0
  3  0  0  7
> rand(partition1$cluster,partition2$cluster,adj=FALSE)
[1] 1
```

L'index de Rand est égal à 1, cela signifie que les deux partitions sont identiques (visible également dans la table de contingence). La classification de type k-means met en évidence la même classification que celle obtenue après une classification hiérarchique.

Ainsi, au regard des résultats de la classification hiérarchique et de la classification k-means , il apparait clairement que les 3 dimensions du concept d'utilisabilité sont des dimensions indépendantes et robustes.

Un point important de nos résultats est que :

- la dimension d'efficacité est dépendante de la condition expérimentale (condition "fixe" versus condition "mobile") ;
- la dimension de satisfaction est dépendante du type d'interface (taille du clavier "petit", "intermédiaire" ou "grande") ;
- alors que la dimension d'efficience ne semble pas dépendre du protocole expérimental mis en jeu mais sûrement de données individuelles (capacité d'un individu à gérer ses ressources cognitives).

5. Conclusion

L'objectif de ce travail était de traiter des données complexes issues de la technique des *Brain Computer Interfaces* (BCI) au moyen de méthodes statistiques multivariées (approche PCAmix et classification

de variables) afin de mieux comprendre et interpréter les relations qui existent entre elles. Les données traitées étaient mixtes (i.e. composées de variables numériques et de variables catégorielles) et permettaient de mesurer les trois dimensions du concept d'utilisabilité : efficacité, efficience et satisfaction. Au niveau expérimental, deux conditions ont été proposées : une condition fixe (pas de mobilité des yeux) et une condition mobile (avec mobilité des yeux) à travers l'interaction de trois claviers virtuels de taille différente. Ces deux conditions "fixe" et "mobile" permettent de simuler deux états de la maladie de Charcot. Un des enjeux importants est de pouvoir faire des préconisations (choix de l'interface) favorisant une meilleure acceptabilité de la part des utilisateurs finaux. Ceci est d'autant plus crucial quand on a affaire à des personnes fragiles comme les personnes souffrants de sclérose latérale amyotrophique (SLA ou maladie de Charcot) par exemple. Ainsi, au regard des résultats de l'approche PCAmix et de ceux de la classification de variables (hiérarchique ou de type k-means), il apparaît clairement que les trois dimensions du concept d'utilisabilité sont des dimensions indépendantes et robustes. Un point important de nos résultats est que :

– la dimension d'efficacité est dépendante de la condition expérimentale (condition "fixe" versus condition "mobile") ;
– la dimension de satisfaction est dépendante du type d'interface (taille du clavier "petit", "intermédiaire" ou "grande") ;
– alors que la dimension d'efficience ne semble pas dépendre du protocole expérimental mis en jeu mais sûrement de données individuelles (capacité d'un individu à gérer ses ressources cognitives).

L'intérêt des approches d'analyse de données et de classification de variables est que ces dernières nous permettent ici de proposer des recommandations concernant le choix d'interface dans l'usage des claviers virtuels notamment pour des personnes avec des problèmes moteurs tels que la SLA. Ces recommandations reposent non seulement sur des critères de performances, d'efficience mais aussi de satisfaction. Ainsi, que ce soit en condition "fixe" ou en condition "mobile", l'interface la plus pertinente, pour la technique des BCIs, notamment en terme de satisfaction est le clavier de taille intermédiaire et ceci à performances équivalentes.

Pour conclure, cette technologie innovatrice de BCI représente un réel outil de communication pour des personnes souffrant d'importantes incapacités motrices. Elle pourrait encore être améliorée grâce à des analyses d'utilisabilité comme c'est le cas de cette étude. Une recherche orientée vers l'amélioration du design de l'Interface, appliquée à l'optimisation du système pour les utilisateurs finaux, doit être un des objectifs à considérer dans le futur.

Remerciements

Les auteurs adressent leurs remerciements au rédacteur en chef et au relecteur pour leurs commentaires et suggestions qui ont conduit à une amélioration substantielle de la version initiale de ce travail. Les auteurs remercient également le projet LICOM (Référence DPI2015-67064-R).

Bibliographie

Allison, B.Z., Pineda, J.A. (2003). ERPs evoked by different matrix sizes : implications for a brain computer interface (BCI) system. *IEEE Trans. Neural Syst. Rehabil. Eng.*, 11, 110-113.

Chavent, M., Kuentz, V., Liquet B., Saracco, J. (2012a), ClustOfVar : An R Package for the Clustering of Variables. *Journal of Statistical Software*, 50, 1-16.

Dhillon, I.S., Marcotte, E.M., Roshan, U. (2003). Diametrical Clustering for identifying anti-correlated gene clusters. *Bioinformatics*, 19, 1612-1619.

Huber, L., Arabie, P. (1985). Comparing partitions. *Journal of classification*, 2 (1), 193-208.

ISO 9241-11. 1998 Ergonomic requirements for office work with visual display terminals (VDTs)-Part 11 : Guidance on usability.

Kececi, H., Degirmenci, Y., Atakay, S. (2006). Habituation and dishabituation of P300. *Cognitive and Behavioral Neurology*, 19(3), 130-134.

Kleih, S., Nijboer, F., Halder, S., Kübler, A. (2010). Motivation modulates the P300 amplitude during brain-computer interface use. *Clinical Neurophysiology*, 121(7), 1023-1031.

Lerman, I.C.(1991). Foundations of the Likelihood Linkage Analysis (LLA) Classification method. *Applied Stochastic Models and Data Analysis*, 7, 63-76.

Li, Y., Bahn, S., Nam, C.S., Lee, J. (2014). Effects of Luminosity Contrast and Stimulus Duration,on User Performance and Preference in a P300- Based Brain-Computer Interface. *International Journal of Human-Computer Interaction*, 30(2), 151-163.

Lu, J., Speier W, Hu, X., Pouratian, N. (2013). The effects of stimulus timing features on P300 speller performance. *Clinical Neurophysiology*, 124(2), 306-314.

Mangun, G.R., Buck, L.A. (1998). Sustained visual spatial attention produces costs and benefits in response time and evoked neural activity. *Neuropsychologia*, 36(3), 189-200.

Marchetti, M., Priftis, K. (2014). Effectiveness of the P3-speller in brain-computer interfaces for amyotrophic lateral sclerosis patients : a systematic review and meta-analysis. *Frontiers in Neuroengineering*, 7, 12-18.

McFarland, D.J., Sarnacki, W.A., Townsend, G., Vaughan, T., Wolpaw, J.R. (2011). The P300-based brain-computer interface (BCI) : Effects of stimulus rate. *Clinical Neurophysiology*, 122(4), 731-737.

Murata, A., Uetake, A. (2001). Evaluation of mental fatigue in human-computer interaction-Analysis using feature parameters extracted from event-related potential. In *10th IEEE International Workshop on Robot and Human Interactive Communication*, 630-635.

Nam, C.S., Li, Y., Johnson, S. (2010). Evaluation of P300-Based Brain-Computer Interface in Real-World Contexts. *International Journal of Human-Computer Interaction*, 6(6), 621-637.

Nielsen, J. (1993). What is Usability ? In *Usability Engineering*, Cambridge, MA : Academic Press, 23-48.

Nielsen, J. (1994). Heuristic evaluation. In Nielsen, J. and Mark, R.L. (Eds.) *Usability Inspection Methods*, New York : John Wiley & Sons.

Polich, J., Kok, A. (1995). Cognitive and biological determinants of P300 : An integrative review. *Biological Psychology*, 41(2), 103-146.

Schalk, G., McFarland, D., Hinterberger, T., Birbaumer, N., Wolpaw, J. (2004). Bci2000 A general-purpose brain-computer interface (BCI) system. *IEEE Transactions on Biomedical Engineering*, 51(6), 1034-1043.

Sellers, E.W., Krusienski, D.J., McFarland, D.J., Vaughan, T.M., Wolpaw, J.R. (2006). A P300 event-related potential brain-computer interface (BCI) : The effects of matrix size and inter stimulus interval on performance. *Biological Psychology*, 73, 242-252.

Vigneau, E., Chen M., Qannari, E.M. (2015). ClustVarLV : An R Package for the Clustering of Variables around Latent Variables. *R Journal*, 7 (2), 134-148.

Vigneau, E., Qannari, E.M. (2003) Clustering of variables around latent components. *Communications in Statistics - Simulation and Computation*, 32, 1131-1150.

