Journal of Statistical Software

October 2019, Volume 91, Issue 10. doi: 10.18637/jss.v091.i10

Beyond Tandem Analysis: Joint Dimension Reduction and Clustering in R

Angelos Markos
Democritus University
of Thrace

Alfonso Iodice D'Enza
Università degli Studi
di Cassino

Michel van de Velden
Erasmus University
Rotterdam

Abstract

We present the R package **clustrd** which implements a class of methods that combine dimension reduction and clustering of continuous or categorical data. In particular, for continuous data, the package contains implementations of factorial K-means and reduced K-means; both methods combine principal component analysis with K-means clustering. For categorical data, the package provides MCA K-means, i-FCB and cluster correspondence analysis, which combine multiple correspondence analysis with K-means. Two examples on real data sets are provided to illustrate the usage of the main functions.

Keywords: dimension reduction, clustering, principal component analysis, multiple correspondence analysis, K-means.

1. Introduction

Cluster analysis aims to find a meaningful allocation of observations to groups that are similar with respect to a set of observed variables. Similarity between observations may be defined in various ways depending on data specificities (e.g., measurement scales) and corresponding distance/similarity measures. If the set of variables used in determining the similarity is large, calculation of dissimilarities may become difficult. To overcome this problem, methods that combine dimension reduction (i.e., reduce the set of variables by either selecting a subset of variables or by using some function to reduce the dimensionality) with cluster analysis have been proposed.

The most popular way of applying dimension reduction and cluster analysis is by simply executing them sequentially. That is, first the original data are transformed using dimension reduction, then cluster analysis is applied to the transformed data. This method is also known as the tandem approach. As part of a sequential (tandem) approach the user can initially apply a dimension reduction technique and then subject the low-dimensional solution to a clustering algorithm.

Some useful R (R Core Team 2019) packages for dimension reduction that implement principal component analysis (PCA), correspondence analysis (CA), multiple correspondence analysis (MCA) and their variants, include **ade4** (Dray and Dufour 2007), **ca** (Nenadić and Greenacre 2007), **CAvariants** (Beh and Lombardo 2014), **FactoMineR** (Lê, Josse, and Husson 2008), **homals** (De Leeuw and Mair 2009), as well as functions `prcomp()` and `princomp()` from the built-in package **stats**. More sophisticated implementations of tandem approaches are available through the packages **FactoClass** (Pardo and Del Campo 2007) and **FactoMineR** (Lê *et al.* 2008). **FactoClass** implements a sequential strategy described in Lebart, Morineau, and Piron (2000). Dimension reduction (PCA or CA/MCA) is first performed, according to the nature of the data, followed by a clustering of the factor scores. The clustering step implements hierarchical clustering (Euclidean distance and Ward's linkage method) followed by K-means, using the cluster centers obtained from the hierarchical algorithm as the initial partition. This is known as the "consolidation" approach (Lebart *et al.* 2000) and can balance the advantages and disadvantages of hierarchical and partitioning methods, especially when the number of objects is large. Moreover, the package calculates the so-called "test-values" (Lebart, Morineau, and Warwick 1984) to facilitate a description of the obtained clusters. A two-dimensional factorial map of the solution is also provided. In a similar fashion, the function `HCPC()` of **FactoMineR** performs Ward's hierarchical clustering on the results from a PCA or CA/MCA. The consolidation approach is also provided as an option. The number of clusters can be automatically determined based on the highest relative loss of within-group inertia. The package also provides a representation of the clusters on the map induced by the first two principal components, as well as a description of the clusters using the function `catdes()`. **FactoMineR** and **FactoClass**, however, do not provide indices for cluster quality assessment. The package **factoextra** (Kassambara and Mundt 2017) extends the functionality of **FactoMineR** providing elegant visualizations of the results of `HCPC()` based on **ggplot2** (Wickham 2016).

Intuitive and straightforward as the tandem approach may be, it may not yield an optimal cluster allocation as the two methods optimize different criteria. Dimension reduction typically aims to retain as much variance as possible in as few dimensions as possible, whereas cluster analysis aims to find similar and dissimilar observations in the data set and allocate the observations accordingly to clusters. This problem is well-known (e.g., Bock 1987; De Soete and Carroll 1994; Van Buuren and Heiser 1989; Vichi and Kiers 2001) and several solutions have been proposed. In this paper, we consider related methods for joint dimension reduction and clustering of continuous and categorical data. In particular, for continuous (or, interval) data we consider reduced K-means (De Soete and Carroll 1994), factorial K-means (Vichi and Kiers 2001) as well as a compromise version of these two methods. For categorical data, cluster correspondence analysis (Van de Velden, Iodice D'Enza, and Palumbo 2017), which, for the analysis of categorical data, is equivalent to GROUPALS (Van Buuren and Heiser 1989), multiple correspondence analysis and K-means (MCA K-means, Hwang, Dillon, and Takane 2006), and iterative factorial clustering of binary variables (i-FCB, Iodice D'Enza and Palumbo 2013) are considered.

Although most of extant joint dimension reduction and cluster analysis methods have been proposed and derived quite a while ago, their popularity, as measured in terms of published applied studies, appears to be limited. One factor that may play a role in this limited use, may be the lack of publicly available software to carry out the analyses. In this paper we present the R package **clustrd** (Markos, Iodice D'Enza, and Van de Velden 2019) that fills this gap by implementing these methods. Note that we do not concern ourselves here with choosing a "best" method. An appraisal of the methods for continuous data can be found in Timmerman, Ceulemans, Kiers, and Vichi (2010), whereas Van de Velden *et al.* (2017) consider the performance of the methods for categorical data.

The outline of the paper is as follows. Sections 2 and 3 provide a brief introduction of joint dimension reduction and clustering methods for continuous and categorical variables, respectively. Section 4 presents an overview of the R package **clustrd** available from the Comprehensive R Archive Network (CRAN) at `https://CRAN.R-project.org/package=clustrd`. Two examples with real data are presented to illustrate the usage of the main functions. Section 5 discusses limitations and possible extensions.

2. Methods for continuous data

Before giving a brief description of the methods included in the **clustrd** package, we introduce some notation that we shall use throughout the paper. Let $\mathbf{X}$ denote a centered and standardized $n \times Q$ data matrix, $\mathbf{B}$ is a $Q \times d$ columnwise orthonormal loadings matrix, i.e., $\mathbf{B}^\top\mathbf{B} = \mathbf{I}_d$, where d is the user supplied dimensionality of the reduced space. Furthermore, $\mathbf{Z}_K$ is the $n \times K$ binary matrix indicating cluster memberships of the n observations into the K clusters. Finally, we use $\mathbf{G}$ to denote the $K \times d$ cluster centroid matrix.

For the analysis of continuous (interval-scaled) data, two alternatives to the sequential tandem approach are commonly distinguished. That is, projection pursuit (Bock 1987) or, equivalently, reduced K-means (RKM) clustering (De Soete and Carroll 1994), and factorial K-means clustering (Vichi and Kiers 2001). Here we first introduce these two methods by presenting their respective objective functions. Then, using a decomposition of the RKM objective function given in Yamamoto and Hwang (2014), we show that the two methods are closely related and only require one algorithm with different parameter settings.

2.1. Reduced K-means

In RKM clustering (De Soete and Carroll 1994), or, equivalently, projection pursuit (Bock 1987) the simultaneous dimension reduction and cluster analysis problem is tackled in such a way that the cluster allocation and dimension reduction maximizes the *between* variance of the clusters in the reduced space. The RKM objective function is

$$\min \phi_{\text{RKM}}\left(\mathbf{B}, \mathbf{Z}_K, \mathbf{G}\right) = \left\|\mathbf{X} - \mathbf{Z}_K\mathbf{G}\mathbf{B}^\top\right\|^2, \tag{1}$$

where $\|\cdot\|$ denotes the Frobenius norm. Following notation introduced by Yamamoto and Hwang (2014), we insert the solution for the cluster means, that is, $\mathbf{G} = \left(\mathbf{Z}_K^\top\mathbf{Z}_K\right)^{-1}\mathbf{Z}_K^\top\mathbf{X}\mathbf{B}$, to obtain

$$\min \phi_{\text{RKM}}\left(\mathbf{B}, \mathbf{Z}_K\right) = \left\|\mathbf{X} - \mathbf{P}\mathbf{X}\mathbf{B}\mathbf{B}^\top\right\|^2, \tag{2}$$

where $\mathbf{P} = \mathbf{Z}_K\left(\mathbf{Z}_K^\top\mathbf{Z}_K\right)^{-1}\mathbf{Z}_K^\top$. This notation will prove to be convenient for showing the relationship between RKM and factorial K-means. Using the projector matrix $\mathbf{P}$ and the trace operator for the sum of diagonal elements of a matrix, we see that

$$\left\|\mathbf{X} - \mathbf{PXBB}^\top\right\|^2 = \operatorname{Tr}\left(\mathbf{X}^\top\mathbf{X}\right) - \operatorname{Tr}\left(\mathbf{B}^\top\mathbf{X}^\top\mathbf{PXB}\right). \tag{3}$$

Furthermore, as minimizing ϕ_{RKM} amounts to maximizing $-\phi_{\mathrm{RKM}}$, we immediately see that in RKM the between cluster variance (i.e., the second term on the right hand side of Equation 3), in the reduced space, is maximized.

2.2. Factorial K-means

Factorial K-means or FKM (Vichi and Kiers 2001) minimizes the *within* variance of the clusters in the reduced space. It can be described as simultaneous K-means clustering with PCA.

The objective funcion for FKM is

$$\min \phi_{\mathrm{FKM}}\left(\mathbf{B}, \mathbf{Z}_K, \mathbf{G}\right) = \left\|\mathbf{XB} - \mathbf{Z}_K\mathbf{G}\right\|^2. \tag{4}$$

Analogue to the procedure followed for RKM, inserting the solution for $\mathbf{G}$ gives

$$\min \phi_{\mathrm{FKM}}\left(\mathbf{B}, \mathbf{Z}_K\right) = \left\|\mathbf{XB} - \mathbf{PXB}\right\|^2. \tag{5}$$

2.3. Reduced K-means and factorial K-means

Yamamoto and Hwang (2014) propose to decompose the RKM objective function in Equation 2 as:

$$\left\|\mathbf{X} - \mathbf{PXBB}^\top\right\|^2 = \left\|\mathbf{X} - \mathbf{XBB}^\top\right\|^2 + \left\|\mathbf{XB} - \mathbf{PXB}\right\|^2. \tag{6}$$

This decomposition shows that RKM can be seen as a compromise of PCA (the first part of the decomposition) and FKM. Rather than assigning equal weights to the two parts, Vichi, Vicari, and Kiers (2019) propose to minimize a convex combination of them. The objective function thus becomes:

$$\min \phi_{\mathrm{ClusPCA}}\left(\mathbf{B}, \mathbf{Z}_K\right) = \alpha\left\|\mathbf{X} - \mathbf{XBB}^\top\right\|^2 + (1-\alpha)\left\|\mathbf{XB} - \mathbf{PXB}\right\|^2. \tag{7}$$

Using the trace operator and collecting terms, we see that minimizing ϕ_{ClusPCA} amounts to maximizing:

$$\operatorname{Tr}\left(\mathbf{B}^\top\mathbf{X}^\top\left((1-\alpha)\mathbf{P} - (1-2\alpha)\mathbf{I}\right)\mathbf{XB}\right). \tag{8}$$

Hence, for known $\mathbf{Z}_K$, the loadings $\mathbf{B}$ can be obtained by taking the eigendecomposition of $\mathbf{X}^\top\left((1-\alpha)\mathbf{P} - (1-2\alpha)\mathbf{I}\right)\mathbf{X}$, and by selecting orthonormal eigenvectors corresponding to the d largest eigenvalues. On the other hand, for known $\mathbf{B}$, only $\operatorname{Tr}\left(\mathbf{B}^\top\mathbf{X}^\top\left((1-\alpha)\mathbf{P}\right)\mathbf{XB}\right)$ from Equation 8 needs to be maximized. This maximization problem is equivalent to a standard K-means clustering objective function applied to $\mathbf{XB}$. Combining these two parts, yields, for given α, the following alternating least-squares algorithm:

1. Generate an initial cluster allocation $\mathbf{Z}_K$ (e.g., by randomly assigning objects to clusters).

2. Find loadings $\mathbf{B}$ by taking the eigendecomposition of $\mathbf{X}^\top\left((1-\alpha)\mathbf{P} - (1-2\alpha)\mathbf{I}\right)\mathbf{X}$

3. Update the cluster allocation $\mathbf{Z}_K$ by applying K-means to the reduced space subject coordinates $\mathbf{XB}$

4. Repeat the procedure (i.e., go back to step 2) using $\mathbf{Z}_K$ for the cluster allocation matrix, until convergence. That is, until $\mathbf{Z}_K$ remains constant.

Note that, for $\alpha = 0.5$, the problem reduces to RKM, and for $\alpha = 0$ to FKM. When $\alpha = 1$ the solution is equivalent to the tandem approach (principal component analysis followed by K-means of the factor scores). The final model selection can be based on theoretical considerations, for instance by deciding a priori that the desired method is to be a compromise between FKM and RKM (i.e, choosing $\alpha = 0.25$), or for instance, right in the middle of PCA and RKM (i.e., choosing $\alpha = 0.75$). Generally, several values of alpha could be evaluated and the most attractive of these (e.g., the one that leads to the most interesting interpretation) could be chosen. For selecting the number of clusters, Timmerman *et al.* (2010) recommend to apply well-established heuristics, such as the Calinski-Harabasz index (Caliński and Harabasz 1974) or the average silhouette width (Rousseeuw 1987).

Empirical and simulation-based examples indicated that RKM and FKM can correctly identify well-separated clusters masked by randomly generated variables, whereas a corresponding sequential (tandem) approach fails (De Soete and Carroll 1994; Vichi and Kiers 2001). A comparison of the performances of RKM and FKM can be found in Timmerman *et al.* (2010). The authors showed evidence that for both FKM and RKM, the cluster membership recovery generally deteriorates with increasing amount of overlap between clusters. RKM fails to recover the clustering of the objects when the data contain much variance in directions orthogonal to the subspace of the data. On the contrary, RKM generally shows a good performance when the majority of the variables reflect the clustering structure and/or the variables are standardized before analysis. In terms of subspace recovery, RKM and FKM appear to complement each other.

3. Methods for categorical data

In this paper, and in the **clustrd** package, we consider the following three methods for categorical data: MCA K-means (Hwang *et al.* 2006), iterative factorial clustering of binary variables (i-FCB, Iodice D'Enza and Palumbo 2013), and cluster correspondence analysis (cluster CA). Below we give a brief description of these methods. For more details on the methods, their relationships as well as an appraisal of their performance in simulated experiments, see Van de Velden *et al.* (2017).

For the analysis of categorical data, some additional notation is necessary. First of all, instead of using the data matrix $\mathbf{X}$ we use a so-called superindicator matrix $\mathbf{Z}$. That is, an n by Q binary matrix, where for each observation the selected categories are coded as ones, and all other elements are zero. Hence, $\mathbf{Z} = [\mathbf{Z}_1, \mathbf{Z}_2, \ldots, \mathbf{Z}_q]$, where $\mathbf{Z}_j$ is an n by p_j indicator matrix for the j-th categorical variable, the number of categorical variables (attributes) is q and $Q = \sum_{j=1}^{q} p_j$. Correspondingly, and analogue to the loadings matrix in the continuous case, we define $\mathbf{B} = \left[\mathbf{B}_1^\top, \mathbf{B}_2^\top, \ldots, \mathbf{B}_q^\top\right]^\top$ as the $Q \times d$ matrix of category quantifications, where $\mathbf{B}_j$ denotes the p_j by d matrix of category quantifications for the jth categorical variable. The standardization of these category quantifications differs depending on the method. Furthermore, we define $\mathbf{Y}$ as an n by d matrix with reduced space coordinates for the observations (i.e., coordinates for the rows of $\mathbf{Z}$). As before, $\mathbf{Z}_K$ denotes the $n \times K$ cluster membership indicator matrix, $\mathbf{G}$ the $K \times d$ cluster centroid matrix.

3.1. Cluster correspondence analysis

Cluster CA can be seen as correspondence analysis applied to the cross-tabulation of the cluster membership and the variable categories, i.e., the cluster by categories contingency matrix. That is,

$$\mathbf{F} = \mathbf{Z}_K^\top \mathbf{Z}. \tag{9}$$

Applying CA to this matrix yields optimal scaling values for rows (clusters) and columns (categories) in such a way that the between cluster variance is a maximum. That is, the clusters are optimally separated with respect to the distributions over the categorical variables. Similarly, and simultaneously, categories with differing distributions over the clusters are optimally separated. However, the cluster memberships are not known and need to be determined by the method as well. It can be shown, that optimal category quantifications (i.e., column coordinates) as well as an optimal cluster allocation can be obtained by iterating between CA of the contingency matrix (9) and by applying K-means cluster analysis to the reduced space coordinates obtained using the CA category quantifications.

The algorithm for cluster CA can be summarized as follows:

1. Generate an initial cluster allocation $\mathbf{Z}_K$ (e.g., by randomly assigning objects to clusters).
2. Find category quantifications $\mathbf{B}$ by applying CA to the contingency matrix $\mathbf{Z}_K^\top \mathbf{Z}$.
3. Calculate coordinates for the individuals (or objects) by averaging the centered scores using the category quantifications from step 1. That is: $\mathbf{Y} = \frac{1}{q}(\mathbf{I} - \mathbf{1}\mathbf{1}^\top/n)\mathbf{Z}\mathbf{B}$.
4. Update $\mathbf{Z}_K$ by applying K-means clustering to $\mathbf{Y}$.
5. Repeat the procedure (i.e., go back to step 2) using $\mathbf{Z}_K$ for the cluster allocation matrix, until convergence. That is, until $\mathbf{Z}_K$ (and hence $\mathbf{Y}$ and $\mathbf{G}$) remain constant.

Since cluster CA amounts to applying CA to the cross-tabulation of cluster membership with variable categories, the coordinates for rows and columns constitute a biplot of clusters and categories (attributes). Hence, projections of the cluster points on attribute vertices, provide approximations to the cluster by attribute associations. For more details on biplots and their interpretation see, e.g., Gower, Lubbe, and Le Roux (2011).

However, given the large differences in dimensionalities (usually relatively few clusters versus many categories) the typical CA normalizations may not lead to similar spread in the row and column points. Consequently, a joint display of the row and column points may not be very informative. Following proposals by Gower, Groenen, and Van de Velden (2010) and Gower *et al.* (2011), Van de Velden *et al.* (2017) propose to multiply the cluster mean points by a constant γ and the categories by its inverse, in such a way that the average squared deviation from the origin is the same in both sets of points. That is,

$$\gamma = \left(\frac{K}{Q} \operatorname{Tr} \mathbf{B}^\top \mathbf{B} / \operatorname{Tr} \mathbf{G}^\top \mathbf{G} \right)^{1/4}, \tag{10}$$

and the matrices with scaled coordinates, $\mathbf{G}_s$ and $\mathbf{B}_s$, become

$$\mathbf{G}_s = \gamma \mathbf{G} \text{ and } \mathbf{B}_s = \frac{1}{\gamma}\mathbf{B}. \tag{11}$$

Use of this scaling parameter is the default option in **clustrd**.

3.2. MCA K-means

Hwang *et al.* (2006) proposed a joint multiple correspondence analysis and K-means method that combines the two objectives using a convex combination. Using the same notation as before, the MCA K-means objective can be formulated as follows:

$$\min \phi_{\text{mcak}}\left(\mathbf{Y}, \mathbf{B}_j, \mathbf{G}, \mathbf{Z}_K\right) = \alpha \frac{1}{q} \sum_{j=1}^{q} \|\mathbf{Y} - \mathbf{Z}_j \mathbf{B}_j\|^2 + (1-\alpha)\|\mathbf{Y} - \mathbf{Z}_K \mathbf{G}\|^2 \tag{12}$$

subject to

$$\mathbf{Y}^\top \mathbf{Y} = \mathbf{I}_d.$$

The weight α is user supplied and controls the importance of the MCA and K-means part. Although the term $1/q$ does not appear in Hwang *et al.* (2006), it is needed to ensure that for $\alpha = 0.5$, the MCA and cluster analysis parts receive equal weights. Hwang *et al.* (2006) derive the following algorithm:

1. Generate an initial cluster allocation $\mathbf{Z}_K$ (e.g., by randomly assigning objects to clusters) and use MCA to obtain an initial solution for $\mathbf{Y}$.

2. Calculate category quantifications and cluster means using: $\mathbf{B}_j = \left(\mathbf{Z}_j^\top \mathbf{Z}_j\right)^{-1} \mathbf{Z}_j^\top \mathbf{Y}$ and $\mathbf{G} = \left(\mathbf{Z}_K^\top \mathbf{Z}_K\right)^{-1} \mathbf{Z}_K^\top \mathbf{Y}$.

3. Update $\mathbf{Y}$ using the eigenequation:
$$\left(\alpha \frac{1}{p} \sum_{j=1}^{p} \mathbf{Z}_j \left(\mathbf{Z}_j^\top \mathbf{Z}_j\right)^{-1} \mathbf{Z}_j^\top + (1-\alpha)\mathbf{Z}_K \left(\mathbf{Z}_K^\top \mathbf{Z}_K\right)^{-1} \mathbf{Z}_K^\top\right) \mathbf{Y} = \mathbf{Y}\mathbf{\Lambda}.$$

4. Update $\mathbf{Z}_K$ by applying the K-means algorithm to $\mathbf{Y}$.

5. Return to step 2 and repeat until convergence.

Note that step 3 of the algorithm may require an eigendecomposition of a very large matrix (i.e., $n \times n$). To tackle this problem, an alternative formulation given in Van de Velden *et al.* (2017) was considered in our implementation of the algorithm. Although not explicitly mentioned in the original paper, unless the matrices $\mathbf{Z}_j$ and $\mathbf{Z}_K$ are centered, the algorithm above yields a trivial solution as its first dimension which needs to be discarded. See Iodice D'Enza, Van de Velden, and Palumbo (2014) for details.

3.3. i-FCB

The i-FCB approach (Iodice D'Enza and Palumbo 2013) consists of iterations between non-symmetric CA (NSCA, Beh and Lombardo 2014) and K-means clustering. First, NSCA is applied to the cluster by categories contingency matrix (9), where the dependent (reference) variable is the cluster membership indicator (i.e., the rows) and the explanatory variables are the q categorical variables. Next, for the cluster assignment, K-means clustering is performed on object scores (sample coordinates) that are obtained using the centered and weighted super indicator matrix, and the category quantifications from the non-symmetric CA. See Iodice D'Enza and Palumbo (2013) for details. Using our notation, the algorithm becomes:

1. Generate an initial cluster allocation $\mathbf{Z}_K$ (e.g., by randomly assigning objects to clusters).
2. Apply non-symmetric CA to the contingency matrix $\mathbf{F} = \mathbf{Z}_K^\top \mathbf{Z}$ and thus obtain a category quantification matrix $\mathbf{B}$.
3. Calculate subject coordinates $\mathbf{Y} = \mathbf{D}_w(\mathbf{I} - \mathbf{1}\mathbf{1}^\top/n)\mathbf{Z}\mathbf{B}$, where $\mathbf{D}_w = diag\left(\mathbf{Z}_K\mathbf{Z}_K^\top\mathbf{1}\right)$, that is, a diagonal matrix whose non-zero elements indicate, for each subject, the size of the cluster to which it belongs.
4. Apply K-means to $\mathbf{Y}$ to update the cluster allocation matrix $\mathbf{Z}_K$ and return to step 2. Repeat until convergence.

In contrast to the algorithms for cluster CA and MCA K-means, the NSCA and K-means problems consecutively solved in the i-FCB algorithm do not correspond to the same objective function. This complicates the assessment of convergence in i-FCB. In **clustrd** the sum of the NSCA and K-means objective functions is considered, and we iterate until this sum is below a small threshold value. If in two subsequent iterations the value increases, we terminate the algorithm and revert to the solution of the penultimate step. In addition to this convergence issue, it should be noted that the K-means procedure in this algorithm is not straightforward as $\mathbf{Y}$ depends on $\mathbf{Z}_K$ through $\mathbf{D}_w$. To overcome this issue, both $\mathbf{Y}$ and $\mathbf{D}_w$ are fixed in Step 4, therefore $\mathbf{Z}_K$ is the only term updated. The new $\mathbf{D}_w$ is used in Step 3 of the next iteration, when $\mathbf{Y}$ is updated.

An extensive simulation study was conducted by Van de Velden *et al.* (2017) to assess to what extent the three aforementioned methods are able to retrieve existing cluster structure in categorical data. In the presence of correlated noise variables, the tandem approach and MCA K-means with $\alpha = 0.5$ resulted in poorer cluster recovery. Cluster correspondence analysis consistently outperformed the other methods, whereas i-FCB performed well only in balanced scenarios, i.e., for equally-sized clusters.

4. Package description and illustrative examples

In this section we illustrate the functionality available in the **clustrd** package through the analysis of two real data sets. All the methods described in Sections 2 and 3 are implemented in the functions `cluspca()` and `clusmca()` for continuous and categorical data, respectively. To facilitate the selection of the most appropriate number of clusters and dimensions, the package provides the function `tuneclus()`. The three functions return an S3 object of class `cluspca`, `clusmca` or `tuneclus`, respectively, for which `plot()`, `print()`, `summary()` and `fitted()` functions are provided. The package also includes five data sets, three of which have been used in the original papers introducing the methods in question. Table 1 shows a summary of the package contents.

Function	Description
`cluspca()`	Methods for continuous data (reduced K-means and factorial K-means).
`clusmca()`	Methods for categorical data (MCA K-means, i-FCB and cluster CA).
`tuneclus()`	Cluster quality assessment for a range of clusters and dimensions.
`plot()`	Plot method for `cluspca`, `clusmca` and `tuneclus` objects.
`print()`	Prints out some key components of `cluspca`, `clusmca` and `tuneclus` objects.
`summary()`	Produces a detailed output for `cluspca`, `clusmca` and `tuneclus` objects.
`fitted()`	Returns a matrix where each observation is replaced by its cluster center (`method` argument is set to `"centers"`) or a vector of cluster membership (`method` argument is set to `"classes"`) for `cluspca`, `clusmca` and `tuneclus` objects.
Data set	**Description**
`bribery`	The data set refers to a collection of 55 articles on bribery cases from central Russian newspapers 1999–2000 (Mirkin 2005).
`cmc`	Data of married women in Indonesia related to their choice of contraceptive method (Dua and Graff 2017).
`diamond`	Data on 308 diamond stones sold in Singapore. The main attributes are diamond weight, color, clarity, certification body and price (Chu 2001).
`hsq`	Data collected through the humor styles questionnaire (HSQ) which assesses four independent ways in which people express and appreciate humor (Van de Velden *et al.* 2017).
`macro`	Data on the short-term macroeconomic performance of national economies of twenty OECD countries in September 1999 (Vichi and Kiers 2001).

Table 1: Summary of **clustrd** package contents.

4.1. Short-term macroeconomic scenario of OECD countries

To demonstrate the `cluspca()` function we provide an application to data describing a short-term macroeconomic scenario. The `macro` data set contains the values of six economic indicators of twenty countries, members of the organization for economic co-operation and development (OECD, Vichi and Kiers 2001): gross domestic product (GDP), leading indicator (LI), unemployment rate (UR), interest rate (IR), trade balance (TB), net national savings (NNS). Values are percentage change from the previous year. The goal of the analysis is to identify classes of similar economies and also understand the relationships within the set of economic indicators.

Four arguments are required as input in `cluspca()`: a matrix or data frame (`data`); the number of clusters (`nclus`); the number of dimensions (`ndim`) and, through the `method` argument, the desired method. As selected method one may either choose RKM or FKM or, in case intermediate weighting is desired, one may replace `method` by a parameter `alpha` that adjusts for the importance of the two parts in (6). The solution amounts to reduced K-means for `alpha` $= 0.5$ and to factorial K-means for `alpha` $= 0$. Note that, when `alpha` $= 1$ the solution is equivalent to tandem analysis; that is PCA followed by K-means clustering of the objects. A description of the available arguments along with the related output, is given in Table 2. After installing **clustrd** from the CRAN, the package and the `macro` data set are loaded:

```
R> library("clustrd")
R> data("macro", package = "clustrd")
```

Arguments	Description
`data`	Data set with metric variables.
`nclus`	Number of clusters.
`ndim`	Dimensionality of the solution.
`method`	Specifies the method. Options are `"RKM"` for reduced K-means and `"FKM"` for factorial K-means (default = `"RKM"`).
`alpha`	Adjusts for the relative importance of RKM and FKM in the objective function; `alpha` = 0.5 leads to reduced K-means, `alpha` = 0 to factorial K-means, and `alpha` = 1 reduces to the tandem approach.
`center`	A logical value indicating whether the variables should be shifted to be zero centered before the analysis takes place (default = `TRUE`).
`scale`	A logical value indicating whether the variables should be scaled to have unit variance before the analysis takes place (default = `TRUE`).
`rotation`	Specifies the method used to rotate the factors. Options are `none` for no rotation, `varimax` for varimax rotation with Kaiser normalization and `promax` for promax rotation (default = `"none"`).
`nstart`	Number of random starts (default = 100).
`smartStart`	If `NULL` then a random cluster membership vector is generated. Alternatively, a cluster membership vector can be provided as a starting solution.
`seed`	An integer that is used as argument by `set.seed()` for offsetting the random number generator when `smartStart = NULL`. The default value is `NULL`.
Output	**Description**
`obscoord`	Object scores (sample coordinates).
`attcoord`	Variable scores (loadings).
`centroid`	Cluster centroids.
`cluster`	Cluster membership.
`criterion`	Optimal value of the objective function.
`size`	The number of objects in each cluster.
`scale`	A copy of `scale` in the return object.
`center`	A copy of `center` in the return object.
`nstart`	A copy of `nstart` in the return object.
`odata`	A copy of `data` in the return object.

Table 2: List of `cluspca()` arguments and output with description.

In this example, reduced K-means was applied to the data with the argument `method` set to `"RKM"`. The number of clusters and the number of dimensions were chosen based on interpretational ease, following Vichi and Kiers (2001) who have previously considered this data set. Therefore, `nclus` was fixed to 3 and `ndim` to 2. To avoid local minima due to the K-means step, 100 random starts were used (`nstart = 100` is the default). Variables were centered and standardized, which is the default setting (`center = TRUE` and `scale = TRUE`). A varimax rotation of the factors was performed to simplify interpretation (`rotation = "varimax"`). The `summary()` method prints out a detailed summary of the RKM solution, including cluster sizes and centroids, variable scores (loadings) on the two dimensions, the sum-of-squares decomposition, the cluster membership vector and the objective criterion value, as shown below.

```
R> outRKM <- cluspca(macro, 3, 2, method = "RKM",
+    rotation = "varimax", seed = 1234)
R> summary(outRKM)


Solution with 3 clusters of sizes 10 (50%), 7 (35%), 3 (15%)
in 2 dimensions. Variables were mean centered and standardized.

Cluster centroids:
            Dim.1   Dim.2
Cluster 1  0.9264  -0.5039
Cluster 2 -1.4344  -0.3536
Cluster 3  0.2589   2.5049

Variable scores:
      Dim.1   Dim.2
GDP -0.7670  0.2123
LI  -0.1150 -0.2175
UR  -0.4271 -0.1109
IR  -0.0201  0.6607
TB  -0.0318 -0.6532
NNS  0.4634  0.1791

Within cluster sum of squares by cluster:
[1] 5.1105 4.2402 1.9681
 (between_SS / total_SS =  80.05 %)

Clustering vector:
  Australia      Canada     Finland      France       Spain      Sweden
          2           2           2           2           2           2
        USA Netherlands      Greece      Mexico    Portugal     Austria
          2           1           3           3           3           1
    Belgium     Denmark     Germany       Italy       Japan      Norway
          1           1           1           1           1           1
Switzerland          UK
          1           1

Objective criterion value: 34.2877

Available output:

 [1] "obscoord"  "attcoord"  "centroid"  "cluster"
 [5] "criterion" "size"      "odata"     "scale"
 [9] "center"    "nstart"
```

The following classification was obtained (clusters are sorted according to size):

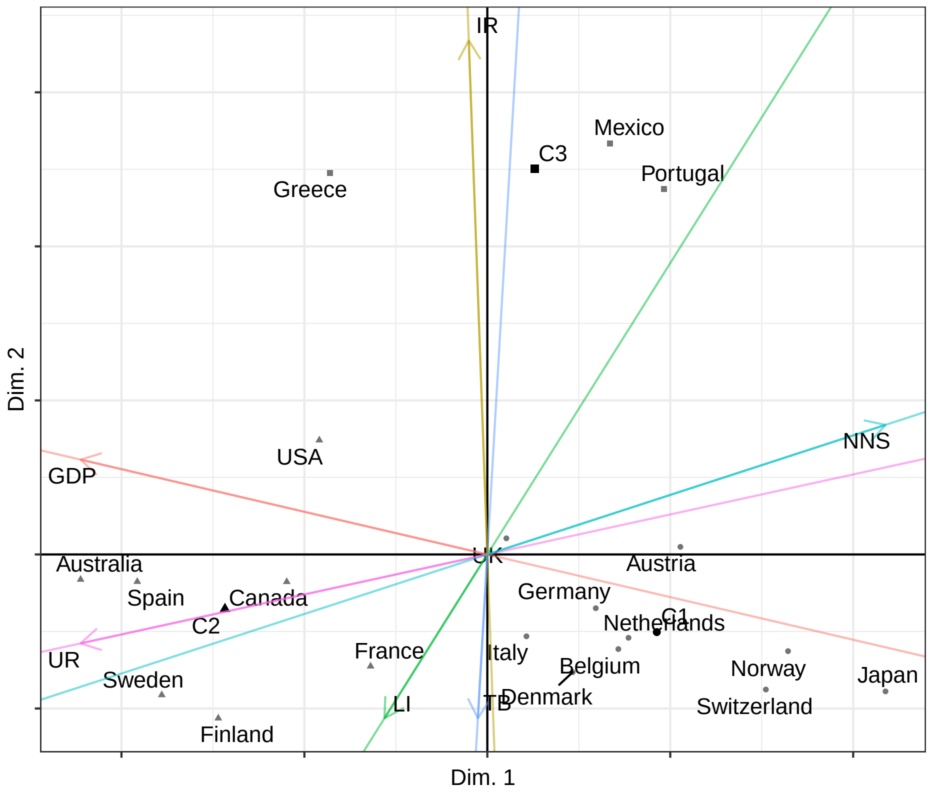

Figure 1: Reduced K-means biplot of countries (points) and economic indicators (biplot axes with respect to components 1 (horizontal) and 2 (vertical). Cluster means are labeled C1 through C3.

- Cluster 1 (50%): Austria, Belgium, Denmark, Germany, Italy, Japan, Netherlands Norway, Switzerland, United Kingdom.
- Cluster 2 (35%): Australia, Canada, Finland, France, Spain, Sweden, United States.
- Cluster 3 (15%): Greece, Mexico, Portugal.

To visualize the solution the generic `plot()` function may be used, which yields a two-dimensional factorial map. The argument `dim` controls which dimensions to plot. The default value is `dim = c(1,2)`, i.e., the first two dimensions are plotted. The argument `what` specifies whether to visualize objects, variables or both; a different map is produced in each case. When `what = c(TRUE, FALSE)` a scatterplot of the objects and cluster centroids is obtained. A correlation circle of the variables is given when `what = c(FALSE, TRUE)`. The default option is `what = c(TRUE, TRUE)` and leads to a biplot where biplot axes (lines with arrows drawn from the origin) are used to represent the variables. A vector of custom attribute labels, `lbl`, is provided via `attlabs = lbl`. Projection of object points onto the biplot axes makes it possible to infer the approximated variable values. The resulting plots are `ggplot2` objects and as such can be stored and customized using standard **ggplot2** functions.

In our working example, the joint display of countries, economic indicators and cluster centroids on the first two dimensions can be obtained via

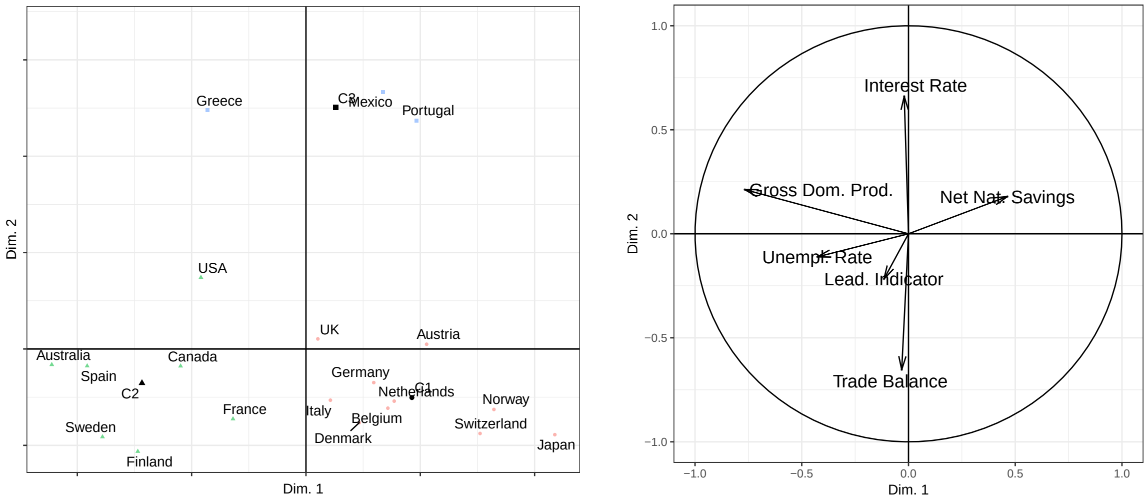

Figure 2: Reduced K-means conventional graphical display with respect to components 1 (horizontal) and 2 (vertical). Left: Scatterplot of the twenty countries; cluster means are labeled C1 through C3. Right: Correlation circle of the six economic indicators.

```
R> plot(outRKM)
```

In the corresponding biplot of Figure 1, countries are represented as points and economic indicators are represented as biplot axes. The first dimension opposes Clusters 1 and 2, whereas the second dimension clearly separates Cluster 3 from the others. The first dimension is characterized mainly by gross domestic product (GDP), unemployment rate (UN) and net national savings (NNS), whereas the second dimension is infuenced mostly by interest rate (IR) and trade balance (TB). A scatterplot of countries and the correlation circle of economic indicators are shown in Figure 2 and can be obtained via

```
R> plot(outRKM, what = c(TRUE, FALSE))
R> lbl <- c("Gross Dom. Prod.", "Lead. Indicator", "Unempl. Rate",
+    "Interest Rate", "Trade Balance", "Net Nat. Savings")
R> plot(outRKM, what = c(FALSE, TRUE), attlabs = lbl)
```

To facilitate a description of the obtained clusters, a parallel coordinate plot of the cluster means can be also provided, by setting the `cludesc` argument to `TRUE`.

```
R> plot(outRKM, cludesc = TRUE)
```

The parallel coordinate plot of Figure 3 depicts the cluster means for each variable and provides quick insights into the difference in growth patterns between groups of countries. Cluster sizes are visualized using thickness of the cluster lines. Notice that the mean values depicted on the vertical axis correspond to mean centered and standardized variables. Countries in the first cluster are mainly characterized by a low growth in GDP, compared to the other clusters. The second cluster contains countries with above average growth in GDP and unemployment rate, and a below average growth in net national savings. Countries in the third group are mainly characterized by a large increase in interest rate and a substantial decrease in trade balance.

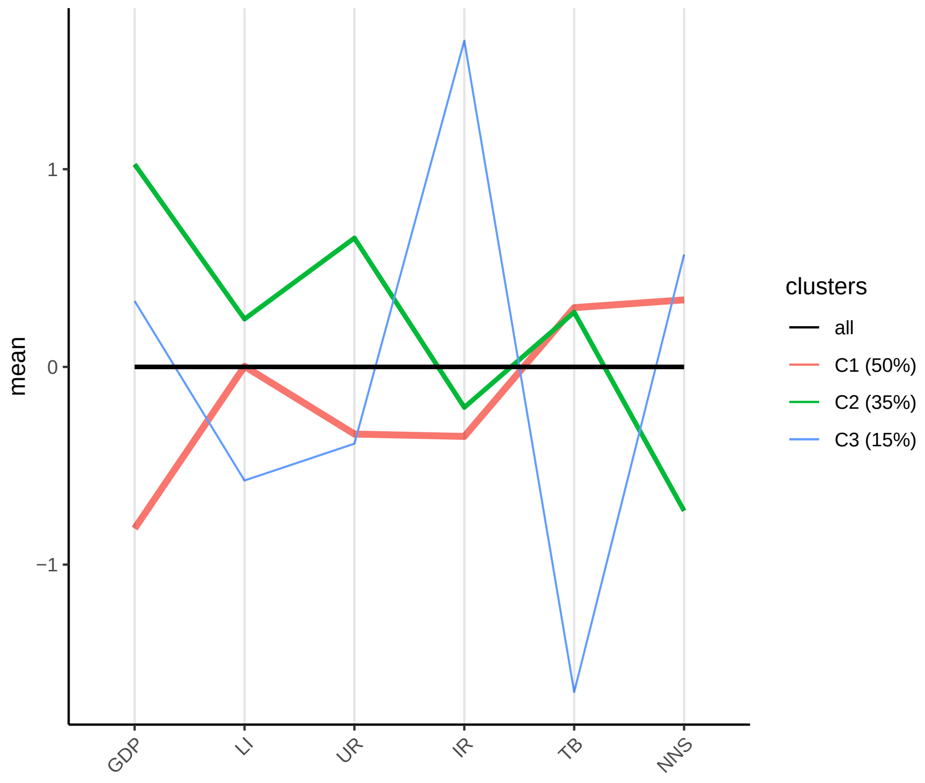

Figure 3: Parallel coordinate plot of the cluster means (line thickness is proportional to cluster size).

4.2. Contraceptive choice in Indonesia

As an illustration of the `clusmca()` function, we consider the `cmc` data set, which is a subset of the 1987 National Indonesia Contraceptive Prevalence Survey. The sample includes 1 473 married women who were not pregnant (or did not know they were pregnant) at the time of the survey. The data set is available through the UCI Machine Learning Repository (Dua and Graff 2017) and is routinely used for a classification task, where the aim is to predict the contraceptive method choice of a woman (no use, long-term methods, or short-term methods), based on nine socio-economic characteristics. In the current context, however, the goal of the analysis is to identify homogeneous groups of Indonesian women, characterized by a small number of socio-economic characteristics and their choice of contraceptive method. The data set includes the following variables:

- `W_AGE`: Wife's age (continuous).
- `W_EDU`: Wife's education (ordinal): `low`, 2, 3, `high`.
- `H_EDU`: Husband's education (ordinal): `low`, 2, 3, `high`.
- `NCHILD`: Number of children (count).
- `W_REL`: Wife's religion (binary): `non-Islam`, `Islam`.

Arguments	Description
`data`	Data set with categorical variables.
`nclus`	Number of clusters.
`ndim`	Dimensionality of the solution.
`method`	Specifies the method. Options are `"MCAk"` for MCA K-means, `"iFCB"` for iterative factorial clustering of binary variables and `"clusCA"` for cluster correspondence analysis (default = `"clusCA"`).
`alphak`	Non-negative scalar to adjust for the relative importance of MCA (`alphak` = 1) and K-means (`alphak` = 0) in the `MCAk` solution (default = .5); `alphak` = 1 reduces to the tandem approach.
`nstart`	Number of random starts (default = 100).
`smartStart`	If `NULL` then a random cluster membership vector is generated. Alternatively, a cluster membership vector can be provided as a starting solution.
`gamma`	Scaling parameter that leads to a similar spread in the object and attribute points (default = `TRUE`).
`seed`	An integer that is used as argument by `set.seed()` for offsetting the random number generator when `smartStart = NULL`. The default value is `NULL`.

Table 3: List of `clusmca()` arguments with description.

- `W_WORK`: Wife's now working? (binary): `Yes`, `No`.
- `H_OCC`: Code of husband's current occupation (categorical): 1, 2, 3, 4.
- `SOL`: Standard-of-living index (ordinal): `low`, 2, 3, `high`.
- `MEDEXP`: Media exposure (binary): `good`, `not good`.
- `CMC`: Contraceptive method used (nominal): `no use`, `long-term`, `short-term`.

The function `clusmca()` requires a matrix or data frame (`data`), the number of clusters (`nclus`), the number of dimensions (`ndim`) and the desired method (`method`). The default method implemented is `"clusCA"`. The solutions of i-FCB and MCA K-means can be obtained by setting the method to `"iFCB"` and `"MCAk"`, respectively. Furthermore, when `method = "MCAk"` it is possible to define the non-negative scalar weight `alphak` that determines the influence of MCA and K-means criteria in the obtained solution. A value of `alphak` closer to 1 puts more weight on the dimension reduction (MCA) part of the objective function, whereas a value closer to 0 gives more importance to the clustering criterion. When `alphak` = 1 the solution is equivalent to tandem analysis; that is MCA followed by K-means clustering of the objects. An investigation of the effects of changing the scalar weight on the final solution can be found in Hwang *et al.* (2006). The description of the available arguments is given in Table 3. The output arguments of `clusmca()` are the same with those described in `cluspca()`.

A cluster CA was conducted on the `cmc` data set, after the values of wife's age and number of children were categorized into three groups based on quartiles, with levels `16-26`, `27-39` and `40-49` years, and `0`, `1-4` and `5 and above`, respectively.

```
R> data("cmc", package = "clustrd")
R> cmc$W_AGE <- ordered(cut(cmc$W_AGE, c(16, 26, 39, 49),
+    include.lowest = TRUE))
```

```
R> levels(cmc$W_AGE) <- c("16-26", "27-39", "40-49")
R> cmc$NCHILD <- ordered(cut(cmc$NCHILD, c(0, 1, 4, 17), right = FALSE))
R> levels(cmc$NCHILD) <- c("0", "1-4", "5 and above")
```

The number of clusters and the number of dimensions were set to 3 and `ndim` to 2, respectively and the algorithm was run with 10 random starts. The `summary()` method returns a similar output to that for `cluspca()` objects (not shown here).

```
R> outclusCA <- clusmca(cmc, 3, 2, nstart = 10, seed = 1234)
R> summary(outclusCA)
R> plot(outclusCA, cludesc = TRUE, topstdres = 20, subplot = TRUE)
```

The cluster CA solution can be visualized with the S3 `plot()` function and returns a two-dimensional factorial map. The default is `dim = c(1, 2)` for plotting the first two dimensions. The cluster centroids are always displayed. This leads to a biplot with the γ-based scaling (see Section 3.1) so as to obtain a similar spread in the object and attribute points (argument `gamma` is `TRUE` by default). To help with the interpretation of clusters it is useful to identify attributes that deviate the most from the independence condition. For this purpose, a series of barplots can be obtained by adding the argument `cludesc = TRUE`. The bars in the resulting plot correspond to the highest (in absolute value) standardized residuals from independence of the attribute distributions conditional to clusters. A positive (negative) residual means that the attribute has an above (below) average frequency within the cluster. The number of top residuals to be plotted is controlled via the argument `topstdres`. The logical argument `subplot` indicates whether a subplot with the full distribution of the standardized residuals will appear at the bottom left corner of the corresponding barplots.

From the corresponding biplot shown in Figure 4, it appears that women in the cluster closest to the center of the plot (i.e., cluster 1 in Figure 4, 45.6%) exhibit characteristics that are closely aligned with the characteristics of the majority in the sample: a moderate education and standard of living and use of short-term contraceptive methods. On the right side of the plot we find a cluster of women (cluster 2 in Figure 4, 41.4%) associated with a high education (both wife and husband), a high standard of living, the 1st category of husband's occupation and a non-Islam religion. Also, women in this cluster are associated with the use of long-term contraceptive methods. Finally, women in the small cluster on the left (cluster 3, 13%) are mostly associated with low education (both wife and husband), not good media exposure and the 4th category of husband's occupation. Note that we do not have any more info on what each category of husband's occupation means.

The three barplots in Figure 5 show for each cluster the twenty attributes with the highest standardized residuals (positive or negative). Figure 5 confirms and enriches the graphical depiction of Figure 4. We observe that in cluster 1, wives and their husbands tend to have moderate levels of education and standard of living, mostly belong to the lowest age category (16–26 years) and tend to use short-term contraceptive methods. In cluster 2, women are characterized by a high education level (wife and husband), the 1st category of husband's occupation, a high standard of living, a non-Islam religion, and mostly reported to be using long-term contraceptive methods. Finally, cluster 3 contains women characterized mostly by a low education (wife and husband), a low standard of living, and not good media exposure. They also tend to belong to the highest age category (40–49 years), had 5 children or above and reported to be using no contraception at all.

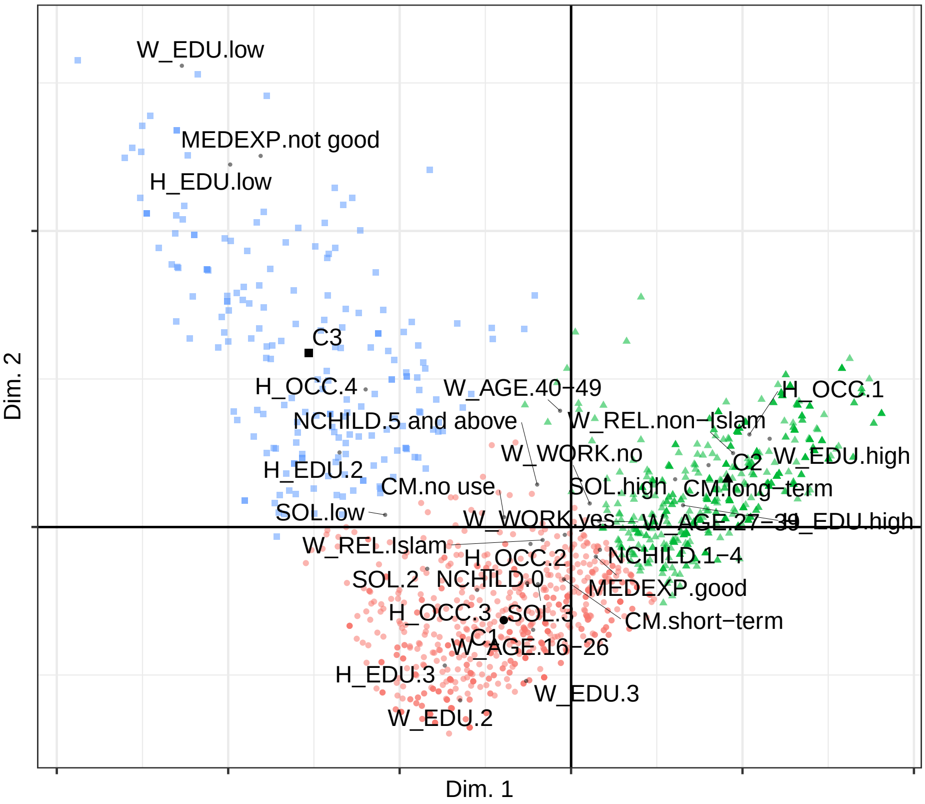

Figure 4: Cluster CA biplot with respect to components 1 (horizontal) and 2 (vertical). Cluster means are labeled C1 through C3.

The selection of the most appropriate number of clusters and dimensions for the `cmc` data set is, as in most clustering problems, not a trivial task and requires a careful assessment of solutions corresponding to different parameter choices. A thorough treatment of these issues is beyond the scope of this paper. However, to facilitate a quantitative appraisal of solutions corresponding to different parameter settings, the **clustrd** package provides the function `tuneclus()`. The function requires a data set (`data` argument), the range of clusters (`nclusrange`) and dimensions (`ndimrange`), as well as the desired method through the `method` argument. The function relies on `cluster.stats()` in the package **fpc** (Hennig 2019) and provides two well established distance-based statistics to assess the quality of an obtained clustering solution via the `criterion` argument; the average silhouette width (ASW) index (`criterion = "asw"`, Rousseeuw 1987) and the Calinski-Harabasz (CH) index (`criterion = "ch"`, Caliński and Harabasz 1974). The ASW index, which ranges from -1 to 1, reflects the compactness of the clusters and indicates whether a cluster structure is well separated or not. The CH index is the ratio of between-cluster variance to within-cluster variance, corrected according to the number of clusters, and takes values between 0 and ∞. In general, the higher the ASW and CH values, the better the cluster separation.

As alternative measure to evaluate the number of clusters, one may also simply consider the objective value (value of the optimization criterion) of the chosen method (`criterion = "crit"`). In particular, by inspecting how the objective value is affected by changes in the

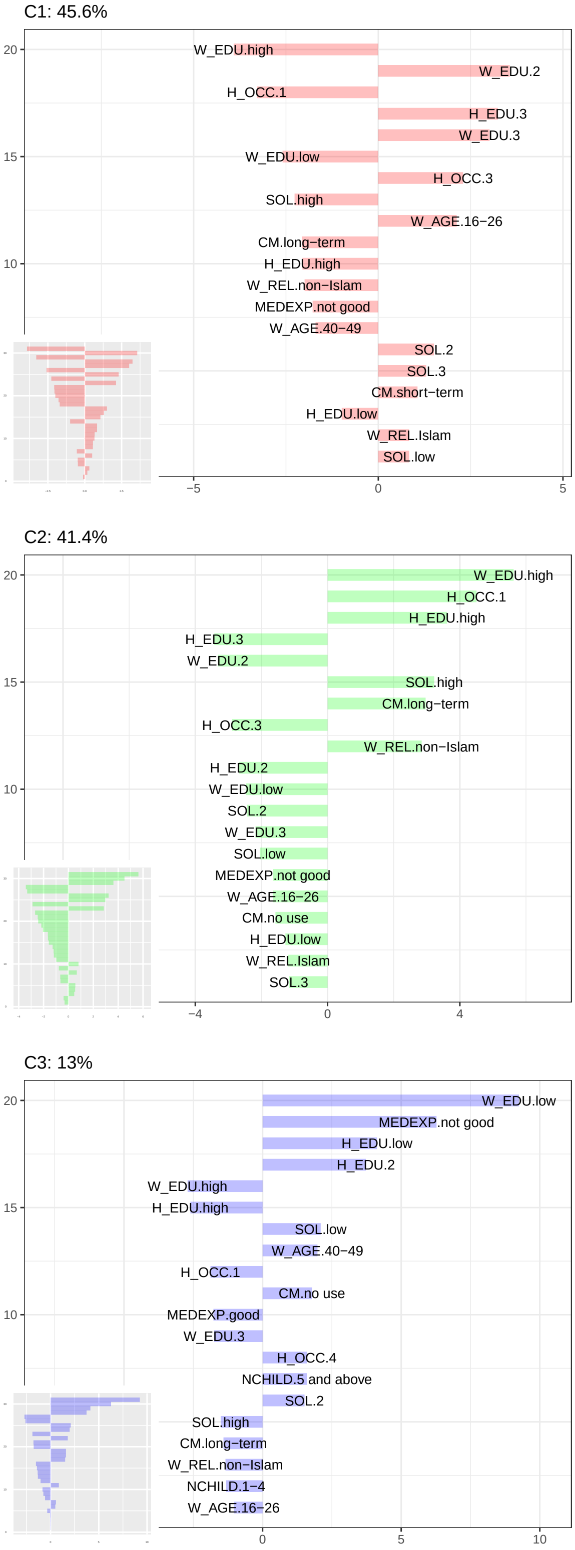

Figure 5: Top 20 of the largest standardized residuals per cluster (with complete distributions in small subplots).

Arguments	Description
`data`	Data set with metric or categorical variables.
`nclusrange`	An integer vector with the range of clusters which are to be compared by a cluster quality criterion.
`ndimrange`	An integer vector with the range of dimensions which are to be compared by a cluster quality criterion.
`method`	Specifies the method. Options are `"RKM"` for reduced K-means, `"FKM"` for factorial K-means, `"MCAk"` for MCA K-means, `"iFCB"` for iterative factorial clustering of binary variables and `"clusCA"` for cluster correspondence analysis.
`criterion`	One of `"asw"`, `"ch"` or `"crit"`. Determines whether the average silhouette width, Calinski-Harabasz index or objective value of the selected method is used (default = `"asw"`).
`dst`	Specifies the data used to compute the distances between objects. Options are `"full"` for the original data (after possible scaling) and `"low"` for the object scores in the low-dimensional space (default = `"full"`).
`...`	Further arguments to be transferred to `cluspca()` or `clusmca()`.
Output	Description
`clusobjbest`	The output of the optimal run of `cluspca()` or `clusmca()`.
`nclusbest`	The optimal number of clusters.
`ndimbest`	The optimal number of dimensions.
`critbest`	The optimal criterion value for `nclusbest` clusters and `ndimbest` dimensions.
`critgrid`	Matrix of size `nclusrange` × `ndimrange` with the criterion values for the specified ranges of clusters and dimensions (values are calculated only when the number of clusters is greater than the number of dimensions; otherwise values in the grid are left blank).

Table 4: List of `tuneclus()` arguments with description.

number of clusters, it may be possible to select an optimal number of clusters. For example, similar to the use of scree plots in PCA, one could search for a number of clusters after which no substantial improvement in the objective value is realized (Hwang *et al.* 2006). It is important to emphasize that the provided measures, that is, the three criteria can only be used to select an optimal number of dimensions and clusters among different solutions of a certain method. Comparison of the values across methods is generally not meaningful.

The function `tuneclus()` allows the specification of the data matrix used to compute the distances between objects, via the `dst` argument. When `dst = "full"` (default) the appropriate distance measure is computed on the original data. In particular, the Euclidean distance can be used for continuous variables and Gower's distance for categorical variables. When the option is set to `"low"`, the distance is computed between the low-dimensional object scores. In that case, distances are affected by the chosen dimension reduction method and the value of the selected cluster quality criterion can be misleading, as the low-dimensional projection forces object points to be close to cluster centroids. For MCA K-means, this is particularly obvious as we can immediately influence our results by changing (decreasing) `alphak`. Consequently, one cannot use this option when comparing results of different methods. On the other hand, conditional on a certain method, the "low" option can be informative concerning the selection of the optimal number of dimensions and clusters.

The output of the tuning function contains the optimal numbers of clusters (nclusbest) and dimensions (ndimbest) based on the chosen criterion value (critbest), the output of the optimal run of cluspca() or clusmca() (clusobjbest), and a grid of size nclus × ndim, with the selected clustering quality criterion values corresponding to the specified ranges of clusters and dimensions (critgrid). Note that these values are calculated for nclus > ndim. Table 4 lists all the available arguments for tuneclus().

To demonstrate parameter tuning, we show an application of MCA K-means to the cmc data for a wide range of clusters (3 to 10) and dimensions (2 to 9). The Euclidean distance was computed between individuals on the original data matrix (dst = "full"). The method argument was set to "MCAk" with 10 random starts (nstart = 10) and the average silhouette width was used as a cluster quality assessment criterion (criterion = "asw").

```
R> bestMCAk <- tuneclus(cmc, 3:10, 2:9, method = "MCAk",
+    criterion = "asw", dst = "full", nstart = 10, seed = 1234)
R> bestMCAk

The best solution was obtained for 3 clusters of sizes 633 (43%),
611 (41.5%),  229 (15.5%) in 2 dimensions, for a cluster quality
criterion value of 0.188.

The average Silhouette width values of each cluster are:
[1] 0.21 0.19 0.12

Cluster quality criterion values across the specified range of clusters
(rows) and dimensions (columns):
       2     3     4     5     6     7     8      9
3  0.188
4  0.119 0.168
5  0.102 0.151 0.075
6  0.075 0.127 0.053 0.136
7  0.075 0.115 0.087 0.047 0.108
8  0.061 0.097 0.075 0.105 0.055 0.114
9  0.038 0.089 0.098 0.089  0.09 0.082 0.025
10 0.043  0.09 0.086 0.073 0.046 0.101 0.104 -0.055

Cluster centroids:
            Dim.1   Dim.2
Cluster 1  0.0291 -0.0116
Cluster 2 -0.0138  0.0292
Cluster 3 -0.0437 -0.0459

Within cluster sum of squares by cluster:
[1] 0.0065 0.0071 0.0054
 (between_SS / total_SS =  99.13 %)
```

```
Objective criterion value: 8.2462

Available output:

[1] "clusobjbest" "nclusbest"   "ndimbest"
[4] "critbest"    "critgrid"    "crit"
[7] "cluasw"
```

As shown in output, the optimal (highest) average silhouette width value (`critbest`) equals 0.188 and was obtained for 3 clusters (`nclusbest`) and 2 dimensions (`ndimbest`), with cluster-specific ASW values of 0.21, 0.19 and 0.12, respectively.

5. Conclusion

In this paper we described the R package **clustrd** that implements a class of methods combining dimension reduction and cluster analysis. There is a variety of packages that provide methods for dimension reduction and/or distance-based clustering. However, joint methods are currently not available in any other software. The **clustrd** package fills this gap. Existing methods and their relationships were briefly presented. These methods have been implemented using two main functions, `cluspca()` and `clusmca()`, for continuous and categorical data, respectively. The **clustrd** package also provides visualizations for the factorial solutions and the description of clusters, as well as a function to decide on the number of clusters and dimensions.

The methods implemented in the **clustrd** are not exhaustive. Recently, Yamamoto and Hwang (2014) proposed a generalization of reduced K-means that accounts for a subset of variables related amongst each other, but unrelated to the cluster structure. Their method, which they call generalized reduced clustering, requires tuning of a parameter related to the dimensionalities of the "cluster related" and the "cluster unrelated" set. If the dimensionality of the subset of variables unrelated to the cluster structure but related amongst each other, is zero, their method corresponds to the problem formulated in (8), where the weights are not required to sum to one. Generalized reduced clustering is not included in **clustrd**.

In future versions of the package, the authors will seek to include fuzzy extensions of the methods currently implemented, such as fuzzy MCA K-means (Hwang, Dillon, and Takane 2010) and two-way regularized fuzzy MCA K-means (Kim, Choi, and Hwang 2017). These approaches typically require the selection of a fuzzy scalar or weight, which controls the fuzziness of the clustering solution.

Extensions of joint dimension reduction and clustering methods for handling mixed data also exist: GROUPALS (Van Buuren and Heiser 1989; Vichi *et al.* 2019). However, the implementation of such methods is not trivial as they often require data specific and subjective choices concerning variable homogenization. For example, categorization of continuous variables or (constrained) optimal scaling of the categorical ones. Such developments might also be included in future versions of the package, after further investigation.

References

Beh EJ, Lombardo R (2014). *Correspondence Analysis: Theory, Practice and New Strategies*. John Wiley & Sons, Chichester. doi:10.1002/9781118762875.

Bock HH (1987). "On the Interface between Cluster Analysis, Principal Component Analysis, and Multidimensional Scaling." In H Bozdogan, AK Gupta (eds.), *Multivariate Statistical Modeling and Data Analysis*, volume 8 of *Theory and Decision Library B – Mathematical and Statistical Methods*, pp. 17–34. Springer-Verlag, Dordrecht. doi:10.1007/978-94-009-3977-6_2.

Caliński T, Harabasz J (1974). "A Dendrite Method for Cluster Analysis." *Communications in Statistics – Theory and Methods*, **3**(1), 1–27. doi:10.1080/03610927408827101.

Chu S (2001). "Pricing the C's of Diamond Stones." *Journal of Statistics Education*, **9**(2). doi:10.1080/10691898.2001.11910659.

De Leeuw J, Mair P (2009). "Gifi Methods for Optimal Scaling in R: The Package **homals**." *Journal of Statistical Software*, **31**(4), 1–20. doi:10.18637/jss.v031.i04.

De Soete G, Carroll JD (1994). "K-Means Clustering in a Low-Dimensional Euclidean Space." In E Diday, Y Lechevallier, M Schader, P Bertrand, B Burtschy (eds.), *New Approaches in Classification and Data Analysis*, Studies in Classification, Data Analysis, and Knowledge Organization, pp. 212–219. Springer-Verlag, Berlin. doi:10.1007/978-3-642-51175-2_24.

Dray S, Dufour AB (2007). "The **ade4** Package: Implementing the Duality Diagram for Ecologists." *Journal of Statistical Software*, **22**(4), 1–20. doi:10.18637/jss.v022.i04.

Dua D, Graff C (2017). "UCI Machine Learning Repository." URL http://archive.ics.uci.edu/ml.

Gower JC, Groenen PJF, Van de Velden M (2010). "Area Biplots." *Journal of Computational and Graphical Statistics*, **19**(1), 46–61. doi:10.1198/jcgs.2010.07134.

Gower JC, Lubbe SG, Le Roux NJ (2011). *Understanding Biplots*. John Wiley & Sons, Chichester. doi:10.1002/9780470973196.

Hennig C (2019). ***fpc**: Flexible Procedures for Clustering*. R package version 2.2-3, URL http://CRAN.R-project.org/package=fpc.

Hwang H, Dillon WR, Takane Y (2006). "An Extension of Multiple Correspondence Analysis for Identifying Heterogenous Subgroups of Respondents." *Psychometrika*, **71**(1), 161–171. doi:10.1007/s11336-004-1173-x.

Hwang H, Dillon WR, Takane Y (2010). "Fuzzy Cluster Multiple Correspondence Analysis." *Behaviormetrika*, **37**(2), 111–133. doi:10.2333/bhmk.37.111.

Iodice D'Enza A, Palumbo F (2013). "Iterative Factor Clustering of Binary Data." *Computational Statistics*, **28**(2), 789–807. doi:10.1007/s00180-012-0329-x.

Iodice D'Enza A, Van de Velden M, Palumbo F (2014). "On Joint Dimension Reduction and Clustering of Categorical Data." In D Vicari, A Okada, G Ragozini, C Weihs (eds.), *Analysis and Modeling of Complex Data in Behavioral and Social Sciences*, Studies in Classification, Data Analysis, and Knowledge Organization, pp. 161–169. Springer-Verlag, Cham. doi:10.1007/978-3-319-06692-9_18.

Kassambara A, Mundt F (2017). ***factoextra**: Extract and Visualize the Results of Multivariate Data Analyses.* R package version 1.0.5, URL http://CRAN.R-project.org/package=factoextra.

Kim S, Choi JY, Hwang H (2017). "Two-Way Regularized Fuzzy Clustering of Multiple Correspondence Analysis." *Multivariate Behavioral Research*, **52**(1), 31–46. doi:10.1080/00273171.2016.1246996.

Lê S, Josse J, Husson F (2008). "**FactoMineR**: An R Package for Multivariate Analysis." *Journal of Statistical Software*, **25**(1), 1–18. doi:10.18637/jss.v025.i01.

Lebart L, Morineau A, Piron M (2000). *Statistique Exploratoire Multidimensionnelle*. Dunod, Paris.

Lebart L, Morineau A, Warwick KM (1984). *Multivariate Descriptive Statistical Analysis: Correspondence Analysis and Related Techniques for Large Matrices*. John Wiley & Sons, New York.

Markos A, Iodice D'Enza A, Van de Velden M (2019). ***clustrd**: Methods for Joint Dimension Reduction and Clustering.* R package version 1.3.6-2, URL https://CRAN.R-project.org/package=clustrd.

Mirkin B (2005). *Clustering for Data Mining: A Data Recovery Approach*. Chapman & Hall/CRC, London.

Nenadić O, Greenacre M (2007). "Correspondence Analysis in R, with Two- And Three-Dimensional Graphics: The **ca** Package." *Journal of Statistical Software*, **20**(3), 1–13. doi:10.18637/jss.v020.i03.

Pardo CE, Del Campo PC (2007). "Combination of Factorial Methods and Cluster Analysis in R: The Package **FactoClass**." *Revista Colombiana de Estadística*, **30**(2), 231–245. doi:10.15446/rce.

R Core Team (2019). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. URL https://www.R-project.org/.

Rousseeuw PJ (1987). "Silhouettes: A Graphical Aid to the Interpretation and Validation of Cluster Analysis." *Journal of Computational and Applied Mathematics*, **20**, 53–65. doi:10.1016/0377-0427(87)90125-7.

Timmerman M, Ceulemans E, Kiers HAL, Vichi M (2010). "Factorial and Reduced K-Means Reconsidered." *Computational Statistics & Data Analysis*, **54**(7), 1858–1871. doi:10.1016/j.csda.2010.02.009.

Van Buuren S, Heiser WL (1989). "Clustering n Objects into k Groups under Optimal Scaling of Variables." *Psychometrika*, **54**, 699–706. doi:10.1007/bf02296404.

Van de Velden M, Iodice D'Enza A, Palumbo F (2017). "Cluster Correspondence Analysis." *Psychometrika*, **82**(1), 158–185. doi:10.1007/s11336-016-9514-0.

Vichi M, Kiers HAL (2001). "Factorial k-Means Analysis for Two-Way Data." *Computational Statistics & Data Analysis*, **37**, 49–64. doi:10.1016/s0167-9473(00)00064-5.

Vichi M, Vicari D, Kiers HAL (2019). "Clustering and Dimension Reduction for Mixed Variables." *Behaviormetrika*, **46**(2), 243–269. doi:10.1007/s41237-018-0068-6.

Wickham H (2016). *ggplot2: Elegant Graphics for Data Analysis*. Springer-Verlag, New York. ISBN 978-3-319-24277-4. URL https://ggplot2.tidyverse.org/.

Yamamoto M, Hwang H (2014). "A General Formulation of Cluster Analysis with Dimension Reduction and Subspace Separation." *Behaviormetrika*, **41**, 115–129. doi:10.2333/bhmk.41.115.

Affiliation:

Angelos Markos
Department of Primary Education
Democritus University of Thrace
68100 Alexandroupoli, Greece
E-mail: amarkos@eled.duth.gr
URL: http://amarkos.gr/

Journal of Statistical Software http://www.jstatsoft.org/
published by the Foundation for Open Access Statistics http://www.foastat.org/

October 2019, Volume 91, Issue 10 *Submitted:* 2016-11-01
doi:10.18637/jss.v091.i10 *Accepted:* 2018-07-03