Statistical Methods & Applications
https://doi.org/10.1007/s10260-020-00546-2

ORIGINAL PAPER

# An empirical comparison of two approaches for CDPCA in high-dimensional data

Adelaide Freitas[1,2] · Eloísa Macedo[3] · Maurizio Vichi[4]

Accepted: 7 August 2020

## Abstract

Modified principal component analysis techniques, specially those yielding sparse solutions, are attractive due to its usefulness for interpretation purposes, in particular, in high-dimensional data sets. Clustering and disjoint principal component analysis (CDPCA) is a constrained PCA that promotes sparsity in the loadings matrix. In particular, CDPCA seeks to describe the data in terms of disjoint (and possibly sparse) components and has, simultaneously, the particularity of identifying clusters of objects. Based on simulated and real gene expression data sets where the number of variables is higher than the number of the objects, we empirically compare the performance of two different heuristic iterative procedures, namely ALS and two-step-SDP algorithms proposed in the specialized literature to perform CDPCA. To avoid possible effect of different variance values among the original variables, all the data was standardized. Although both procedures perform well, numerical tests highlight two main features that distinguish their performance, in particular related to the two-step-SDP algorithm: it provides faster results than ALS and, since it employs a clustering procedure (k-means) on the variables, outperforms ALS algorithm in recovering the true variable partitioning unveiled by the generated data sets. Overall, both procedures produce satisfactory results in terms of solution precision, where ALS performs better, and in recovering the true object clusters, in which two-step-SDP outperforms ALS approach for data sets with lower sample size and more structure complexity (i.e., error level in the CDPCA model). The proportion of explained variance by the components estimated by both algorithms is affected by the data structure complexity (higher error level, the lower variance) and presents similar values for the two algorithms, except for data sets with two object clusters where the two-step-SDP approach yields higher variance. Moreover, experimental tests suggest that the two-step-SDP approach, in general, presents more ability to recover the true number of object clusters, while the ALS algorithm is better in terms of quality of object clustering with more homogeneous, compact and well-separated clusters in the reduced space of the CDPCA components.

**Electronic supplementary material** The online version of this article (https://doi.org/10.1007/s10260-020-00546-2) contains supplementary material, which is available to authorized users.

Extended author information available on the last page of the article

Published online: 18 August 2020

**Keywords** Principal component analysis · Clustering of objects · Partitioning of attributes · Semidefinite programming

## 1 Introduction

Ever-increasing problem size demands the development of novel techniques to perform statistical analysis. Clustering and dimensionality reduction techniques have been widely studied and applied on many real-life multivariate data and in various scientific areas, such as machine learning, pattern recognition, engineering, bioinformatics and image processing (Xu and Wunsch 2005). Clustering aims to find a meaningful assignment of objects into groups that are similar w.r.t. a set of issues or rules previously established. Dimensionality reduction based on principal component analysis (PCA) is aimed at representing a high-dimensional data into a lower dimensional space, retaining the maximum variability of the original attributes. This projection to a low-dimensional space is provided by a new set of attributes called principal components (PCs), which are uncorrelated and defined by linear combinations of the original attributes (Jolliffe 2002). Typically, the coefficients of these linear combinations, called (unit-)loadings by some authors, are nonzero, which may be cumbersome or even a shortcoming for the interpretation of the PCs. This is particularly relevant in computational biology where the number of variables is, in general, very large. In an attempt to construct more interpretable PCs, PCA-based methodologies providing components with zero loadings have been proposed in the literature. Regardless of the sparseness constraints, either ensuring the sparseness in each component loadings or, less restrictive, in the loadings matrix (Adachi and Trendafilov 2016), high computational complexity is present in those methodologies. The simple PCA restricts the component loadings to be − 1, 0 or 1 (Vines 2000). The maximal variance approach SCoTLASS, proposed by Jolliffe et al. (2003), introduces a bound on the sum of absolute values of the loadings, and some become zero. Modified PCs with sparse loadings have been also constructed using, for instance, the LASSO (elastic net) regression method (Zou et al. 2006), convex semidefinite programming (SDP) relaxations (d'Aspremont et al. 2007), a variable projection solver (Erichson et al. 2018), and an iterative thresholding approach (Ma 2013).

There exist also PCA approaches that involve partitioning of attributes, providing disjoint components which are expected to be of great importance for interpretation purposes. It should be noticed that disjoint components lead to a sparsity level of, at least, 50% for the loading matrix since, in this case, the number of nonzero elements in the loading matrix coincides with the number of variables. In this sense, approaches yielding disjoint components can be considered as sparse PCA. In Enki et al. (2013), the so-called interpretable PCs is proposed which are based on nonoverlapping components constructed from the correlation matrices of clustered attributes, maximizing the explained variance. In Vichi and Saporta (2009), a constrained PCA called clustering and disjoint principal component analysis (CDPCA) is proposed which partitions objects into nonoverlapping homogeneous clusters and, simultaneously, finds disjoint components with maximum variance, such that the between cluster deviance in the reduced space of the components is maximized. This simultaneous action of CDPCA on objects and variables overcomes the drawback of the so-called “tandem analysis” where two reduction techniques are applied sequentially, in particular, when the reduction of the objects is obtained by applying a clustering method on the score matrix resulting from a PCA procedure using the first few principal components, which may mask the clustering structure of the data (DeSarbo et al. 1990). CDPCA is a two-mode methodology aimed to providing the clustering of objects on the reduced set of the CDPCA components. Basically, CDPCA intents to describe the data matrix by a reduced set of object centroids which were identified by k-means and a set of disjoint PCs provided by the application of PCA on the set of centroids. Recently, a joint graphical representation of both the samples and the variables of a data matrix using the HJ-Biplot method with disjoint factorial axes were constructed based on the CDPCA methodology (Nieto-Librero et al. 2017).

In this paper, we focus our attention on CDPCA applied on high-dimensional data namely, when the number of variables is much greater than the number of objects. We briefly review two recently proposed iterative heuristic procedures based on two steps for performing CDPCA on two-way data (Macedo 2015; Macedo and Freitas 2015), both following the idea of the four-step alternating least-squares (ALS) algorithm proposed in Vichi and Saporta (2009). One of these procedures was presented in Macedo and Freitas (2015), herein simply called ALS, and corresponds to a two-step version of the original ALS, involving one step for the allocation of objects via k-means and another step for the reduction of the attribute space via application of PCA on the resulting centroids. The other procedure uses an approximation algorithmic framework based on a semidefinite programming (SDP) approach and was called two-step-SDP (Macedo 2015), because two SDP problems are considered for clustering objects and attributes. This latter process can also be subdivided into two phases, involving a first phase where the clusters of objects and attributes are initially estimated using projections and SDP models, and a second phase where a rounding procedure based on k-means applied in the reduced space of centroids is executed in order to refine solutions. Both algorithms, ALS and two-step-SDP, were implemented and tested in R (Development Core 2019) on several data sets (Macedo 2015).

Since the way the data decomposition parameters are estimated is different for the ALS and two-step-SDP algorithms, the purpose of this study is to compare the results obtained by the application of these two approaches and to check whether they provide substantially different outcomes when high-dimensional data are fitted by CDPCA models. Additionally, outcomes provided by the standard and other sparse PCA and the k-means technique are subject to comparison. This paper contributes to the empirical literature on CDPCA in the sense that features of the CDPCA components, for a different number of attribute clusters, and features of the obtained object clusterings, in the reduced space of these components, are being investigated on high-dimensional data sets.

The paper is organized as follows. In Sect. 2, we provide a brief overview of the CDPCA methodology, detailing the model behind each approach, ALS and two-step-SDP. A description of the algorithms ALS and two-step-SDP is included and some major differences between them are highlighted. In Sect. 2.5, specific details of the R implementations `CDpca` and `TwostepSDPClust` are referred. Section 3 presents numerical experiments with the ALS and two-step-SDP algorithms applied on three real gene expression data sets with different number of classes of objects. An empirical comparison of performances is also presented. The main concluding remarks are made in Sect. 4.

## 2 A general overview of the CDPCA methodology

Given a $(I \times J)$ real data matrix $\mathbf{X} = \left[x_{ij}\right]$, the main idea of CDPCA methodology is to cluster the $I$ objects into $P$ nonempty and nonoverlapping clusters $C_p, p = 1, \dots, P$, which are identified by theirs centroids, and, simultaneously, to partitioning the $J$ attributes into $Q$ disjoint components, $PC_q, q = 1, \dots, Q$. The assignment of objects into $P$ clusters and the assignment of attributes into $Q$ components can be stored in binary matrices, $\mathbf{U} = \left[u_{ip}\right]_{I\times P}$ and $\mathbf{V} = \left[v_{jq}\right]_{J\times Q}$, respectively, where

$$u_{ip} = \begin{cases} 1, \text{ if object } & i \in C_p \\ 0, \text{ otherwise} & \end{cases} \quad \text{and} \quad v_{jq} = \begin{cases} 1, \text{ if attribute } & j \in PC_q \\ 0, \text{ otherwise} & \end{cases} \tag{1}$$

The $(P \times J)$ object cluster centroid matrix is $\bar{\mathbf{X}} = \left(\mathbf{U}^T\mathbf{U}\right)^{-1}\mathbf{U}^T\mathbf{X}$, where each row corresponds to its cluster centroid.

### 2.1 The CDPCA model

The CDPCA model describes the data matrix $\mathbf{X}$ as a result of applying PCA to the transformed data matrix $\mathbf{U}\bar{\mathbf{X}}$ where each original object of $\mathbf{X}$ was replaced by its cluster centroid obtained from applying the k-means algorithm to the original data matrix $\mathbf{X}$. Hence, the data matrix $\mathbf{X}$ would be fitted by the CDPCA model as follows (Vichi and Saporta 2009):

$$\begin{aligned} \mathbf{X} =& \mathbf{U}\bar{\mathbf{X}} + \mathbf{E}_1 \quad (\text{k-means on } \mathbf{X}) \\ =& \mathbf{U}\bar{\mathbf{Y}}\mathbf{A}^T + \mathbf{E}_1 + \mathbf{E}_2 \quad (\text{PCA on } \mathbf{U}\bar{\mathbf{X}}) \\ =& \mathbf{U}\bar{\mathbf{Y}}\mathbf{A}^T + \mathbf{E} \quad (\text{CDPCA model}) \end{aligned} \tag{2}$$

where $\mathbf{A}$ is the $(J \times Q)$ component loading matrix having a similar structure of $\mathbf{V}$, with the nonzero elements on each column replaced by the loadings of each component, $\bar{\mathbf{Y}} := \bar{\mathbf{X}}\mathbf{A}$ is a $(P \times Q)$ object centroid matrix in the reduced space of the components and $\mathbf{E} = \mathbf{E}_1 + \mathbf{E}_2$ with $\mathbf{E}_1, \mathbf{E}_2$ the $(I \times J)$ error matrices arising from k-means and PCA, respectively.

The parameters $\mathbf{U}, \bar{\mathbf{Y}}$ and $\mathbf{A}$ can be estimated by minimizing the error associated to the model, that equivalently corresponds to maximizing the term non-associated to the error in the model (2) (Vichi and Saporta 2009). So, the CDPCA problem can be formulated as

$$\max_{\mathbf{U},\bar{\mathbf{Y}},\mathbf{A}} \|\mathbf{U}\bar{\mathbf{Y}}\mathbf{A}^T\|_2^2, \tag{3}$$

subject to **U** being a binary and row stochastic matrix and **A** a columnwise orthonormal matrix, i.e., $\mathbf{A}^T\mathbf{A} = \mathbf{I}$, where each row contributes to a single column (Vichi and Saporta 2009). Due to the binary characteristic of the assignment matrices **U** and **V**, the optimization problem (3) for obtaining the CDPCA decomposition is quite difficult to solve.

Moreover, the CDPCA is considered a constrained PCA methodology by Vichi and Saporta (2009), because it lies on several constraints on the optimization problem. It should be mentioned that this is slightly different of what is suggested in e.g., Hunter and Takane (2002) and Takane and Hunter (2001), since it is defined that a constrained PCA incorporates auxiliary information (internal and external) about the rows and columns of the data matrix. In CDPCA case, we have a constrained optimization problem in the form (3) and subject to specific constraints on the elements of the object assignment matrix $\mathbf{U} = [u_{ip}]$ and the loading matrix $\mathbf{A} = [a_{jq}]$ instead of the data matrix as follow

$$u_{ip} \in \{0, 1\}, \quad i = 1, 2, \dots, I, \quad p = 1, 2, \dots, P,$$
$$\sum_{p=1}^{P} u_{ip}^2 = 1, \quad i = 1, 2, \dots, I,$$
$$\sum_{j=1}^{J} a_{jq}^2 = 1, \quad q = 1, 2, \dots, Q,$$

and

$$\sum_{j=1}^{J} \left(a_{jq}a_{jr}\right)^2 = 0, \quad q = 1, 2, \dots, Q-1, \quad r = q+1, \dots, Q.$$

First group of two constraints ensures that each object belongs to a single group. The last group of two constraints ensures the columnwise orthonormal property of the matrix **A** and the disjoint property of the principal components in CDPCA.

## 2.2 Two-step algorithms for CDPCA

### 2.2.1 ALS

Since $\mathbf{A}^T\mathbf{A} = \mathbf{I}$ and $\bar{\mathbf{Y}} = \bar{\mathbf{X}}\mathbf{A}$, then the objective function in the CDPCA problem (3) can be written in terms of $\bar{\mathbf{X}}$. Indeed,

$$\begin{aligned}\|\mathbf{U}\bar{\mathbf{Y}}\mathbf{A}^T\|_2^2 =& \mathrm{tr}\left(\mathbf{U}\bar{\mathbf{Y}}\mathbf{A}^T\mathbf{A}\bar{\mathbf{Y}}^T\mathbf{U}^T\right) = \mathrm{tr}\Big(\mathbf{U}\bar{\mathbf{Y}}\left(\mathbf{U}\bar{\mathbf{Y}}\right)^T\Big)\\ =&\|\mathbf{U}\bar{\mathbf{Y}}\|_2^2 = \|\mathbf{U}\bar{\mathbf{X}}\mathbf{A}\|_2^2,\end{aligned}$$

which represents the between (object) cluster deviance in the reduced space of the components. It follows that the problem (3) can be rewritten as

$$\max_{\mathbf{U},\bar{\mathbf{X}},\mathbf{A}} \|\mathbf{U}\bar{\mathbf{X}}\mathbf{A}\|_2^2, \tag{4}$$

In order to solve the problem (4), an alternating least-squares (ALS) algorithm, designed with four steps, was proposed in Vichi and Saporta (2009). Recently, in Macedo and Freitas (2015), we showed that each iteration of the ALS algorithm can be summarily described by two basic steps: the assignment of objects via k-means, and the reduction of the attribute space via application of PCA to the resulting centroids. This simplified version of the ALS algorithm is detailed in Macedo and Freitas (2015).

It is worth mentioning that, at the end of each iteration of ALS, a new loading matrix **A** is constructed and the (finite) objective function in (3) is updated. If the objective function increases with this uploading then the previous matrix of the loadings **A** is changed by this new one, otherwise the previous matrix **A** is preserved for the next iteration of the algorithm. This strategy guarantees the convergence of the sequence of objective function values to a stationary point, which is expected to be, at least, a local maximum of the problem (3), as mentioned in Vichi and Saporta (2009). This procedure can be considered as a heuristic and thus, to increase the possibility to achieve the global maximum, it has been suggested by Vichi and Saporta (2009) to run the algorithm several times for different initial assignment matrices **U** and **V**, randomly chosen at the beginning of each run.

### 2.2.2 Two-step-SDP

More recently, in Macedo (2015), a new algorithm that combines semidefinite programming (SDP) models and the CDPCA methodology was proposed, motivated by Macedo and Freitas 2015, Peng and Wei 2007, Peng and Xia 2005, and Vichi and Saporta 2009. It was called two-step-SDP, since two clustering problems are considered, one for object and another for attribute partitions, both solved using SDP models.

Defining $\mathbf{Z} := \mathbf{U}(\mathbf{U}^T\mathbf{U})^{-1}\mathbf{U}^T$ the orthogonal projection matrix onto the column space of the assignment matrix **U**, then,

$$\begin{aligned}\|\mathbf{U}\bar{\mathbf{Y}}\mathbf{A}^T\|_2^2 =& \|\mathbf{Z}\mathbf{X}\mathbf{A}\mathbf{A}^T\|_2^2 = \mathrm{tr}\left(\mathbf{Z}\mathbf{X}\mathbf{A}\mathbf{A}^T\mathbf{A}\mathbf{A}^T\mathbf{X}^T\mathbf{Z}^T\right)\\ =&\mathrm{tr}\left(\mathbf{Z}\mathbf{X}\mathbf{A}(\mathbf{Z}\mathbf{X}\mathbf{A})^T\right) = \|\mathbf{Z}\mathbf{X}\mathbf{A}\|_2^2.\end{aligned}$$

Thus, the optimization problem (3) can be then formulated as

$$\max_{\mathbf{Z},\mathbf{A}} \|\mathbf{Z}\mathbf{X}\mathbf{A}\|_2^2 \tag{5}$$

subject to **Z** being the orthogonal projection matrix onto the column space of **U**, and **A** a columnwise orthonormal matrix. Being **Z** a projection matrix, it satisfies $\mathbf{Z}^2 = \mathbf{Z}$ and $\mathbf{Z} = \mathbf{Z}^T$, and, therefore, **Z** is positive semidefinite.

It is shown in Macedo (2015) that the problem (5) can be solved using two separate clustering problems, each one reformulated in the form of a nonlinear SDP-based model: one for clustering objects with $\mathbf{Z}$ as variable, and another for partitioning attributes with the orthogonal projection matrix onto the column space of the assignment matrix $\mathbf{V}$, i.e., the matrix $\mathbf{H} := \mathbf{V}(\mathbf{V}^T\mathbf{V})^{-1}\mathbf{V}^T$, as variable that presents the same properties of matrix $\mathbf{Z}$. The objective function on each of the clustering problems is $\text{tr}(\mathbf{X}\mathbf{X}^T\mathbf{Z})$ and $\text{tr}(\mathbf{X}^T\mathbf{X}\mathbf{H})$, where tr is the trace of a matrix, respectively, for clustering objects and clustering attributes (Macedo 2015). Since these two clustering models ended up to be 0–1 SDP problems, relaxation has to be considered. Such clustering problems are then solved using a SDP-based approximation algorithmic framework (Macedo 2015): first, a SDP relaxed model is obtained and solved using a procedure based on the characterization of the sum of the largest eigenvalues of a symmetric matrix introduced in Overton and Womersley (1993), which provides almost optimal solutions (Macedo 2015); then, a rounding procedure is used to extract a feasible solution for the clustering of objects and attributes by performing the k-means algorithm in the reduced space of the components. Detailed of the two-step-SDP algorithm can be found in Macedo (2015).

### 2.3 ALS versus two-step-SDP

The ALS and two-step-SDP algorithms start from an initialization step and use iterative schemes in the estimation of the matrices $\mathbf{U}, \mathbf{V}$ and $\mathbf{A}$. Both algorithm can be considered heuristic procedures.

The initialization step in ALS uses random matrices for $\mathbf{U}$ and $\mathbf{V}$, while in the two-step-SDP algorithm, a SDP-based approximation algorithm introduced in Peng and Wei (2007) and Peng and Xia (2005) is used to construct the initial matrices $\mathbf{U}$ and $\mathbf{V}$. This last fact is an advantage of the two-step-SDP algorithm in terms of computation time and sensitivity to the initial solutions, since these are expected to be solutions close to the optimal assignment (Macedo 2015). In contrast, based on numerical experiences on data sets with $I > J$, for the ALS algorithm has been recommended to run it at least 30 times, for different initial assignment matrices $\mathbf{U}$ and $\mathbf{V}$ randomly chosen at the beginning of each run, in order to increase the chance of finding the global optimum of (4), and consequently, to reduce the sensitivity of the ALS algorithm on the initial matrices $\mathbf{U}$ and $\mathbf{V}$ (Vichi and Saporta 2009). The ALS algorithm ensures an improvement of the solution in each iteration (Vichi and Saporta 2009).

In the ALS algorithm, in each iterative step, the matrices $\mathbf{V}$ and $\mathbf{A}$ are updated using an alternating procedure working row-by-row and column-by-column on these matrices, so that components obtained by PCA explain the largest variance, maximizing the between cluster deviance in the reduced space. Such alternating procedure has to be performed $JQ$ times at each iteration, and thus, it may be quite expensive in terms of computation time and memory. Both matrices are simultaneously estimated.

In the two-step-SDP algorithm, the (*almost* optimal) initial $\mathbf{U}$ and $\mathbf{V}$ are refined at each iteration of the rounding procedure where the update of $\mathbf{A}$ requires only one step. Here, the dimensionality reduction is done by finding a partition of the attributes specified by the matrix $\mathbf{V}$, and then, the component loadings specified in $\mathbf{A}$ are computed for this particular partition. It follows that the dimensionality reduction provided by the two-step-SDP algorithm may not explain the largest variance.

Concerning the partition of the $J$ variables in $Q$ components, in the ALS algorithm, this is defined by the $Q$ disjoint components which are obtained from an iterative procedure involving PCA method. Therefore, the partitioning of variables in the CDPCA model estimated by the ALS algorithm may not be appropriate to perceiving the clustering structure of the variable in the data set, but for selecting groups of original variables that provide higher explained variances. In opposite, when the parameters of the CDPCA model are estimated using the two-step-SDP approach, the partitioning of variables can capture a clustering structure since, at the end of the iterative procedure, a clustering method (k-means) in the reduced space of the components is performed to get groups of variables forming the variable partition. This means that, concerning the partitioning of the variables, those two algorithms have different purposes, and hence, may produce different solutions for the problem (3).

Both algorithms, ALS and two-step-SDP, yield disjoint (and possibly sparse) principal components, and which should make the interpretability of the components an easier task.

### 2.4 The choice of the numbers *P* and *Q*

Regardless the algorithm used, ALS or two-step-SDP, the choice of suitable number of both clusters of objects ($P$) and attributes ($Q$) is a preliminary step in any application of the CDPCA methodology. Although this issue is not the core of the present research, based on Rocci and Vichi (2008) a statistic to find an acceptable pair ($P$, $Q$) is briefly discussed which needs further investigation yet. For standard clustering techniques (one-mode methodologies), the selection of the number of clusters is not an easy task and is, in general, based on monitoring the behaviour of a few internal clustering validation measures (e.g., average silhouette width, Dunn index) [see, for instance, Charrad et al. (2014)]. For a two-mode methodology, Rocci and Vichi (2008) proposed to use the Calinski and Harabasz' criterion (Calinski and Harabasz 1974) to select, simultaneously, the number of clusters of objects and the number of components. Therefore, herein, a similar index based on a pseudo-F statistic and given as follows might be suggested:

$$pF = \frac{\|\mathbf{U}\bar{\mathbf{Y}}\mathbf{A}^T\|_2^2/\, df_B}{\|\mathbf{X} - \mathbf{U}\bar{\mathbf{Y}}\mathbf{A}^T\|_2^2/df_W},$$

where $df_B$ ($df_W$) denotes the degrees of freedom of the squared sum $\|\mathbf{U}\bar{\mathbf{Y}}\mathbf{A}^T\|_2^2$ ($\|\mathbf{X} - \mathbf{U}\bar{\mathbf{Y}}\mathbf{A}^T\|_2^2$, resp.) which corresponds to the reconstructed (by $\mathbf{Y} = \mathbf{A}\mathbf{X}$) between-class (within-class, resp.) deviance of the partition given by $\mathbf{U}$ of the data matrix $\mathbf{X}$ (Vichi and Saporta 2009). The number $df_B$ is the number of free elements that are necessary to estimate to calculate

$$\|\mathbf{U}\bar{\mathbf{Y}}\mathbf{A}^T\|_2^2 = \sum_{i=1}^{I}\sum_{j=1}^{J}\left(\sum_{p=1}^{P}\sum_{q=1}^{Q} u_{ip}\bar{y}_{pq}a_{jq}\right)^2$$

The matrix $\bar{\mathbf{Y}} = [\bar{y}_{pq}]$ has $P \times Q$ elements without constraints. The loadings matrix $\mathbf{A}$ has $J \times Q$ elements, but there are $J(Q-1)+Q$ constraints: by row ($J$ rows) there are $Q-1$ elements that should be equal to zero, and by column ($Q$ columns) there is one constraint, since each column of $\mathbf{A}$ has unit norm. Therefore,

$$df_B = PQ + JQ - (J(Q-1)+Q) = PQ + J - Q$$

Since $I \times J$ is the dimension of the data matrix, the degree of freedom of $\|\mathbf{X}\|_2^2$ is given by $IJ - 1$. Finally, since

$$\|\mathbf{X}\|_2^2 = \|\mathbf{X} - \mathbf{U}\bar{\mathbf{Y}}\mathbf{A}^T\|_2^2 + \|\mathbf{U}\bar{\mathbf{Y}}\mathbf{A}^T\|_2^2$$

[see Vichi and Saporta (2009) for more details], then $df_W = IJ - (PQ + J - Q)$.

The combination of numbers ($P$, $Q$) corresponding to the highest value of $pF$ should be chosen, since it indicates where there is low within-class and high between-class deviance. However, as emphasized in Rocci and Vichi (2008), the pseudo-$F$ statistic $pF$ may not give reliable indications when there are not quite distinct clusters. In practical situations, clusters completely distinguishable may be difficult to occur and a data-driven guide to adjust the selection of $P$ and $Q$, provided by a complementary analysis of characteristics of the elements belonging to each object and variables clusters, might be useful.

Still, it may also happen that, for instance, due to the nature of the data, one of the numbers, $P$ or $Q$, is known (e.g., for gene expression data sets, the samples may have been extracted from known groups of diseases, or the genes have been selected by a gene ontology level defining each group by the genes belonging to the same biological functional process). In this situation, we can proceed in a standard way by calculating internal clustering validation indexes (e.g., Dunn index). For instance, having prior knowledge on $Q$ and using Dunn index, this index is calculated for different values of $P$, according to partitions provided by applications of CDPCA with $P$ clusters of objects. Then, the number $P$ that provides the highest observed value of the Dunn index should be selected.

## 2.5 Software

We have computationally developed two functions in the open-source software `R` (Development Core 2019) to apply CDPCA on standardized data using the above approaches, namely the function `CDpca` available in the R-based package biplotbootGUI (Nieto-Librero et al. 2019) for performing ALS, and the function `TwostepSDPClust`, available in Supplementary Material, for performing two-step-SDP. The previous normalization of the data is aimed at avoiding the influence of different variance values among the variables in the definition of the components (similarly to the standard PCA method). Both functions are user friendly with few arguments, as described in Table 1. Besides these standard arguments, the user

**Table 1** Input arguments common to our implemented R-functions CDpca and TwostepSDPClust

| Input | Description |
|---|---|
| data | A $(I \times J)$ numeric data matrix |
| P | The number of clusters of objects |
| Q | The number of clusters of attributes |
| class | A vector of length $I$ containing an integer code identifying the true classification of objects, or NULL and 0 if no classification is known for CDpca and TwostepSDPClust, respectively |
| tol | A small convergence tolerance value |
| maxit | The maximum number of iterations |

should also specify the number of runs (r) of the ALS algorithm for the final solution by the CDpca function. Both functions return estimated parameters for the CDPCA model: $\mathbf{U}$, $\mathbf{A}$, and $\bar{\mathbf{Y}}$.

## 3 Performance of CDPCA in high-dimensional data

The performance and efficiency of the aforementioned algorithms, implemented in the R functions CDpca and TwostepSDPClust, were evaluated through an experimental comparative study involving simulated and three real gene expression data sets. The goals of our simulation studies were to get a better insight into the two procedures according to inherent complexity of data structure and to evaluate their sensitivity to the initial solutions randomly generated as input parameters in both algorithms. Using real data sets, our aim was to conduct a more detailed comparison of results when the ALS and the two-step-SDP algorithms are applied in CDPCA modelling. Notice that since both algorithms are executed on standardized data, the disjoint components will be constructed considering the correlation structure induced by such normalization.

### 3.1 Description of the computational experiments

The numerical experiments were carried out on a computer with an Intel Core i7-2630QM processor CPU@2.0 GHz, with Windows 7 (64 bits) and 12GB RAM, using R version 3.2.5. The input arguments tol, maxit and r of the R-functions mentioned in Sect. 2.5 were empirically chosen. Concretely, the tolerance value (tol) was set to $10^{-8}$ for the function TwostepSDPClust, and $10^{-5}$ for CDpca. The maximum number of iterations (maxit) was set to 1000 for both functions. For the function CDpca, to decrease the chance of missing the global optimum of (4), all the numerical tests were executed with r $= 20$, with the exception of one data set (*SRBCT*, see below), where only r $= 10$ runs were performed, because the computation time of the ALS algorithm increased in an unreasonable way in this case. This particular choice of input arguments leaded to the stability of our numerical results.

To assess the quality of the overall fit of the CDPCA model estimated by the algorithms ALS and two-step-SDP, and their sensitivity to the starting random solutions, some of the following measures were considered: the sum of the squared residuals of the model, the proportion of the variance explained by the components, the level of non-sparsity, and, in the reduced space of the components, the proportion of total variation between centroids of object clusters, i.e., the proportion of the between cluster deviance (bcd). To examine the quality of the object clustering provided by the final estimated CDPCA model, widely used clustering validation indexes were calculated using the R-function `cluster.stats` from the `fpc` package (Hennig 2015): adjusted Rand index (ARI), Melia's variation of information (VI), average silhouette width (ASW), and Dunn index. To evaluate the quality of the disjoint principal components induced by the data correlation structure, estimated by ALS and two-step-SDP algorithms, in recovering the true partition of the original variables, the ARI was measured on simulated data sets where the true partitioning of the variables was known. To analyze the computational efficiency, the running times of both algorithms were compared.

For the three real data sets analyzed in this study, the quality of the sparse principal components rendered by ALS and two-step-SDP algorithms in CDPCA methodology were also comparatively examined with outcomes of a different sparse PCA method (already available in R). Concretely, we chose the function `robspca` of the package `sparsepca` since it executes a (robust) sparse PCA which does not require that users previously know the level of sparsity of each principal component (Erichson et al. 2018). For simplicity, this sparse PCA approach will be herein referred by robSPCA. The proportion of the variance explained by PCA and robSPCA components and their sparsity were subject to comparison. To complement the analysis, the quality of the clustering obtained from CDPCA was compared with that obtained by two different procedures: k-means applied on the original attribute space and k-means applied on the reduced space of robSPCA components (which will depend on the number $Q$ of components considered). Since k-means depends on an initial random solution, for these two clustering procedures, the values of clustering validation measures correspond to their average calculated over 100 runs. The value of coefficient of variation (cv) was also calculated.

## 3.2 Simulation study

Analogous to Vichi (2017), the simulation study carried out in the present work was based on data matrices generated such that model (2) is satisfied. These data are standardized when the ALS and two-step-SDP algorithms were executed. The performance of both algorithms was assessed using the following six evaluation measures: the execution time, bcd, the squared Frobeniuos norm of the residual matrix in the CDPCA model (2) for the standardized data matrix ($||E||_2^2$), the proportion of variance explained by the first two CDPCA components (Var(%)), and the degree of agreement between the true and the estimated groups of both objects and standardized variables using the ARI.

### 3.2.1 Design

Data sets $\mathbf{X}$ were generated using the CDPCA model (2) and satisfying the following conditions:

- the binary matrices $\mathbf{U}$ and $\mathbf{V}$ were randomly generated;
- the object centroid matrix $\bar{\mathbf{Y}} = [\mathbf{y}_1 \dots \mathbf{y}_Q]$ with $\mathbf{y}_q \frown N_P(\mathbf{0}, 30^2\mathbf{I}_P)$;
- the loading matrix $\mathbf{A}$ coincides with $\mathbf{B} \times \mathbf{V}$ with each column orthonormalized, where the matrix $\mathbf{B}$ contains positive and negative values around the value 0.7, namely, $\mathbf{B} = diag(b_1, \dots, b_J)$, with $b_j = 0.7 \times sign(\beta) + 0.05\beta$ and $\beta \frown N(0,1)$; and
- the error matrix $\mathbf{E} = \left[\mathbf{e}_1 \dots \mathbf{e}_J\right]$ with $\mathbf{e}_j \frown N_I(\mathbf{0}, \epsilon^2\mathbf{I}_I)$, where the constant $\epsilon$ represents the error level of the CDPCA model on which the data was generated; the higher $\epsilon$, the more complex the data structure.

Different settings have been considered in the simulations: number of objects $I = 10, 40$ (small sample size), number of object clusters $P = 2, 3, 4$, number of disjoint components $Q = 2, 3, 4$, and three error levels: $\epsilon = 0.1, 1, 2$ corresponding to low, moderate and high complexity of the data structure, respectively. The meaning of the error levels is analogous to that illustrated in Cavicchia et al. (2020) using heatmaps now on both the data matrix and the correlation matrix: the object clusters and the variable partitioning are visibly more evident when a small error $\epsilon$ in the noise $\mathbf{E}$ is considered and tend to be less visible as $\epsilon$ grows. The same high number of variables $J = 1000$ ($\gg I$) was adopted in all settings. Thus, similar ratios $I/J$ ($= 0.01, 0.04$) to those of the three real data sets considered in Sect. 3.3 were analyzed in the simulation studies. A few different numbers for these parameters had been chosen due to high computational effort involved in the execution of the ALS algorithm.

### 3.2.2 Numerical results

To assess the performance of both algorithms, a simulation study was carried out taking a total of 42 setting combinations in which each algorithm was applied on several simulated data sets. Concretely, for each setting, we generated 30 data sets $\mathbf{X}$ as mentioned above. For each generated data set, both algorithms were executed with random initial solutions and, each one of the six measures indicated above were calculated. The values displayed in Tables 2 and 3 are the average of those quantities over the 30 data sets generated in each setting.

Some patterns on the simulation results can be observed in Tables 2 and 3. Clearly, as expected, the two-step-SDP algorithm is faster than ALS algorithm in obtaining the estimated parameters of the CDPCA model. The strategies of both algorithms perform well in providing the maximization of the objective function of the CDPCA problem (3), with the ALS algorithm tending to yield higher deviance between the object clusters identified by itself than for the two-step-SDP case. The goodness of fit of the CDPCA model estimated by both algorithms shows to be similar ($||E||^2_{2,ALS} \approx ||E||^2_{2,SDP}$), though the ALS algorithm performs slightly better in

**Table 2** Average of six evaluation measures over randomly generated 30 data sets by setting with $I = 10$ and $J = 1000$

| (P, Q) | $\epsilon$ | | Time | bcd | $\|\|E\|\|_2^2$ | Var(%) | ARI | |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | Objects | Variables |
| (2, 2) | 0.1 | ALS | 19.07 | 99.99 | 2.35 | 95.67 | 1.00 | – 0.00 |
| | | SDP | 5.42 | 99.86 | 2.35 | 95.70 | 1.00 | 0.56 |
| | 1 | ALS | 13.93 | 98.78 | 7.68 | 44.03 | 0.84 | 0.00 |
| | | SDP | 3.69 | 98.20 | 7.65 | 45.21 | 0.88 | 0.51 |
| | 2 | ALS | 10.45 | 94.06 | 9.04 | 21.28 | 0.53 | 0.00 |
| | | SDP | 5.63 | 96.01 | 8.88 | 24.29 | 0.93 | 0.31 |
| (2, 3) | 0.1 | ALS | 28.61 | 99.97 | 2.70 | 67.90 | 1.00 | 0.00 |
| | | SDP | 5.58 | 99.15 | 2.70 | 79.69 | 1.00 | 0.40 |
| | 1 | ALS | 25.56 | 95.74 | 7.40 | 30.04 | 0.94 | 0.00 |
| | | SDP | 5.62 | 91.29 | 7.38 | 39.22 | 0.97 | 0.34 |
| | 2 | ALS | 18.24 | 93.67 | 8.69 | 19.39 | 0.72 | 0.00 |
| | | SDP | 5.71 | 92.92 | 8.53 | 27.47 | 1.00 | 0.26 |
| (2, 4) | 0.1 | ALS | 35.59 | 99.98 | 2.48 | 53.73 | 1.00 | 0.00 |
| | | SDP | 5.19 | 99.33 | 2.48 | 77.39 | 1.00 | 0.38 |
| | 1 | ALS | 34.15 | 99.52 | 6.97 | 29.64 | 0.99 | 0.00 |
| | | SDP | 5.12 | 94.58 | 6.96 | 45.04 | 1.00 | 0.38 |
| | 2 | ALS | 24.70 | 95.23 | 8.62 | 15.27 | 0.76 | 0.00 |
| | | SDP | 5.32 | 89.92 | 8.51 | 25.99 | 1.00 | 0.20 |
| (3, 2) | 0.1 | ALS | 41.44 | 99.99 | 1.72 | 97.58 | 1.00 | 0.89 |
| | | SDP | 4.09 | 92.57 | 2.76 | 97.44 | 0.84 | 0.96 |
| | 1 | ALS | 68.05 | 99.71 | 6.70 | 60.50 | 0.93 | 0.47 |
| | | SDP | 5.27 | 98.13 | 6.86 | 59.01 | 0.89 | 0.76 |
| | 2 | ALS | 84.61 | 99.17 | 8.19 | 36.55 | 0.74 | 0.11 |
| | | SDP | 5.45 | 98.19 | 8.36 | 33.60 | 0.81 | 0.49 |
| (3, 3) | 0.1 | ALS | 91.52 | 99.99 | 1.30 | 77.69 | 1.00 | 0.88 |
| | | SDP | 5.46 | 99.99 | 1.30 | 76.51 | 1.00 | 0.91 |
| | 1 | ALS | 166.60 | 99.74 | 6.39 | 51.69 | 0.96 | 0.32 |
| | | SDP | 7.61 | 97.96 | 6.58 | 51.60 | 0.98 | 0.58 |
| | 2 | ALS | 179.47 | 98.29 | 7.93 | 41.73 | 0.74 | 0.12 |
| | | SDP | 9.51 | 94.27 | 8.09 | 39.57 | 0.87 | 0.37 |
| (3, 4) | 0.1 | ALS | 133.46 | 99.99 | 1.20 | 62.75 | 1.00 | 0.83 |
| | | SDP | 6.23 | 99.19 | 1.47 | 60.49 | 0.97 | 0.91 |
| | 1 | ALS | 233.32 | 99.82 | 5.87 | 48.83 | 1.00 | 0.57 |
| | | SDP | 10.58 | 97.33 | 6.23 | 47.57 | 0.92 | 0.61 |
| | 2 | ALS | 241.20 | 98.37 | 7.70 | 29.83 | 0.81 | 0.13 |
| | | SDP | 15.79 | 94.69 | 7.98 | 30.87 | 0.91 | 0.35 |

settings with higher number of object clusters ($P = 3, 4$). The complexity of the data structure influences the proportion of variance explained by the first two CDPCA

**Table 3** Average of six evaluation measures over randomly generated 30 data sets by setting with $I = 40$ and $J = 1000$

| Scenario | | | Time | bcd | $\|\|E\|\|_2^2$ | Var(%) | ARI | |
|---|---|---|---|---|---|---|---|---|
| (P, Q) | $\epsilon$ | | | | | | Objects | Variables |
| (2, 2) | 0.1 | ALS | 79.60 | 99.95 | 1.41 | 89.11 | 1.00 | 0.00 |
| | | SDP | 5.42 | 99.64 | 1.41 | 89.24 | 1.00 | 0.60 |
| | 1 | ALS | 79.47 | 98.63 | 4.03 | 34.46 | 0.94 | 0.00 |
| | | SDP | 5.52 | 98.80 | 4.03 | 34.76 | 0.97 | 0.71 |
| | 2 | ALS | 69.73 | 97.32 | 4.61 | 15.26 | 0.73 | 0.00 |
| | | SDP | 5.42 | 96.25 | 4.60 | 16.01 | 0.98 | 0.54 |
| (2, 3) | 0.1 | ALS | 112.02 | 99.98 | 1.31 | 63.41 | 1.00 | 0.00 |
| | | SDP | 5.44 | 99.88 | 1.31 | 75.10 | 1.00 | 0.53 |
| | 1 | ALS | 113.96 | 99.62 | 3.82 | 28.69 | 1.00 | 0.00 |
| | | SDP | 5.40 | 98.60 | 3.82 | 37.02 | 1.00 | 0.63 |
| | 2 | ALS | 112.02 | 98.42 | 4.40 | 15.77 | 0.94 | 0.00 |
| | | SDP | 5.45 | 95.04 | 4.40 | 20.68 | 0.97 | 0.40 |
| (2, 4) | 0.1 | ALS | 128.92 | 99.98 | 1.28 | 48.61 | 1.00 | 0.00 |
| | | SDP | 5.29 | 99.78 | 1.28 | 71.58 | 1.00 | 0.39 |
| | 1 | ALS | 133.31 | 99.52 | 3.75 | 23.11 | 1.00 | 0.00 |
| | | SDP | 5.29 | 97.17 | 3.75 | 35.55 | 1.00 | 0.52 |
| | 2 | ALS | 134.31 | 98.07 | 4.40 | 12.32 | 0.97 | 0.00 |
| | | SDP | 5.37 | 93.65 | 4.39 | 19.80 | 1.00 | 0.42 |
| (3, 2) | 0.1 | ALS | 144.10 | 99.98 | 0.91 | 97.23 | 0.99 | 0.94 |
| | | SDP | 5.55 | 91.27 | 1.49 | 97.18 | 0.87 | 0.98 |
| | 1 | ALS | 219.12 | 99.47 | 3.68 | 46.21 | 0.97 | 0.58 |
| | | SDP | 5.61 | 99.97 | 3.69 | 46.05 | 0.93 | 0.83 |
| | 2 | ALS | 312.84 | 98.71 | 4.42 | 22.41 | 0.86 | 0.30 |
| | | SDP | 5.79 | 97.33 | 4.44 | 22.10 | 0.87 | 0.70 |
| (3, 3) | 0.1 | ALS | 291.03 | 99.99 | 0.72 | 69.44 | 1.00 | 0.89 |
| | | SDP | 7.79 | 93.02 | 1.32 | 68.81 | 0.86 | 0.94 |
| | 1 | ALS | 498.07 | 99.16 | 3.49 | 41.47 | 0.93 | 0.33 |
| | | SDP | 9.35 | 96.55 | 3.56 | 41.14 | 0.91 | 0.87 |
| | 2 | ALS | 626.92 | 98.89 | 4.28 | 22.21 | 0.94 | 0.40 |
| | | SDP | 6.61 | 97.09 | 4.38 | 21.76 | 0.81 | 0.58 |
| (3, 4) | 0.1 | ALS | 359.84 | 99.99 | 0.57 | 53.94 | 1.00 | 0.96 |
| | | SDP | 5.54 | 98.52 | 0.67 | 53.29 | 0.98 | 0.99 |
| | 1 | ALS | 741.08 | 99.69 | 3.27 | 37.80 | 1.00 | 0.57 |
| | | SDP | 8.95 | 95.87 | 3.38 | 38.03 | 0.93 | 0.85 |
| | 2 | ALS | 849.39 | 99.03 | 4.05 | 24.38 | 0.98 | 0.34 |
| | | SDP | 20.87 | 89.51 | 4.10 | 25.13 | 0.93 | 0.69 |

**Table 3** (continued)

| Scenario | | | Time | bcd | $\|\|E\|\|_2^2$ | Var(%) | ARI | |
|---|---|---|---|---|---|---|---|---|
| (P, Q) | $\epsilon$ | | | | | | Objects | Variables |
| (4, 2) | 0.1 | ALS | 140.58 | 99.99 | 0.56 | 98.86 | 1.00 | 1.00 |
| | | SDP | 5.56 | 98.57 | 0.73 | 99.86 | 0.93 | 1.00 |
| | 1 | ALS | 246.91 | 99.84 | 3.32 | 56.48 | 0.98 | 0.82 |
| | | SDP | 5.46 | 97.02 | 3.38 | 56.16 | 0.86 | 0.99 |
| | 2 | ALS | 354.97 | 99.13 | 4.32 | 25.94 | 0.92 | 0.62 |
| | | SDP | 6.65 | 95.41 | 4.36 | 25.58 | 0.82 | 0.84 |
| (4, 3) | 0.1 | ALS | 243.01 | 99.91 | 0.58 | 69.34 | 0.98 | 1.00 |
| | | SDP | 5.50 | 96.62 | 0.91 | 69.34 | 0.87 | 1.00 |
| | 1 | ALS | 382.71 | 99.89 | 3.09 | 48.36 | 1.00 | 0.87 |
| | | SDP | 5.44 | 93.31 | 3.27 | 47.78 | 0.86 | 1.00 |
| | 2 | ALS | 898.04 | 99.25 | 4.22 | 23.32 | 0.97 | 0.51 |
| | | SDP | 7.39 | 97.08 | 4.26 | 23.31 | 0.90 | 0.79 |

components obtained by both algorithms: the proportion decreases when the error level ($\epsilon$) increases. Astonishingly, while the dimensionality reduction provided by the two-step-SDP algorithm may not explain the largest variance, this property seems to be innocuous in data sets with objects splitted in two clusters. In fact, in Tables 2 and 3, the proportion of variance explained by the first two CDPCA components is higher when these components are estimated by two-step-SDP algorithm in data sets with two clusters of objects ($\text{Var}(\%)_{SDP} > \text{Var}(\%)_{ALS}$, for $P = 2$).

Concerning the ability of both algorithms in recovering the true object partition, both algorithms exhibit similar performance. However, the complexity of the data structure seems to influence this ability when the sample size is lower ($I = 10$), in particular the two-step-SDP algorithm performs better in data sets having more complex structure ($\epsilon = 2$), while the ALS algorithm in data sets with lower error level (see Table 2).

The ability of both algorithms to recover the true variable partitioning is quite different. Clearly, the two-step-SDP algorithm performs better in recovering the true variable partition under all the simulated data structure ($\text{ARI}_{SDP} > \text{ARI}_{ALS}$ for variable partition, in Tables 2, 3). Taking into account the algorithmic steps of the two-step-SDP procedure, this fact may be not odd and might be explained as follows. The (final) partitioning of the variables, obtained by the two-step-SDP algorithm, is given by performing a clustering algorithm (k-means), and therefore, a clustering of variables would be expected. Otherwise, since the partitioning of variables by ALS algorithm is obtained via PCA, applied several times during an extensive iterative process, the CDPCA components will be constructed taking into account the highest explained variance. Thus, under ALS procedure, CDPCA may be considered as a variable selection technique where the original variables included in the first CDPCA components might be selected for describing the data.

Furthermore, in both algorithms, the ability for recovering the variable partitioning is affected by the shape of the data. In fact, in Tables 2 and 3, for simulated data sets with a higher number of object clusters ($P = 3, 4$) and having less complex structure ($\epsilon = 0.1$), both algorithms show similar ability in recovering of the true variable partition, exhibiting considerable quality (ARI $\geq 0.88$) and remaining stable when the sample size increases ($I = 10, 40$); however, increasing the error level ($\epsilon$), both algorithms tend to reduce their ability in recovering the true partition. In settings with lower number of object clusters ($P = 2$), the simulation results revealed that the ALS algorithm is not appropriate for the detection of the true variable partition ($\text{ARI}_{ALS} \approx 0$).

Another issue that deserve further investigation using simulations and explored in this study was the sensitivity of ALS and two-step-SDP algorithms to the random initial solutions $\mathbf{U}$ and $\mathbf{V}$. We carried out other simulation study where each algorithm, with random starting solutions, was applied on the same data set several times. Concretely, we considered six different settings as depicted on Table 4, which combine low and moderate error levels ($\epsilon = 0.1, 1$): two settings with $I = 10, P = Q = 2$; two others ones with $I = 40, P = Q = 2$; and more two with $I = 40$ and $P = Q = 3$. For each setting, we have generated 10 data sets $\mathbf{X}$ as mentioned above. For each generated data set, we executed 30 times each algorithm starting with randomly generated initial solutions, and then, we calculated the mean and the standard deviation for the same evaluation measures previously used. The ten values obtained (mean and standard deviation) by measure were synthesized by the arithmetic mean. In Table 4 are depicted the averaged values of the standard deviation to evaluate the variability of the results of both algorithm on the same data set (non-aggregated results for each generated data set are available in Supplementary Material). Low or very low values for the averaged standard deviations of the evaluation measures exhibited on Table 4 demonstrate that the quality of the solutions provided by both algorithms is not affected by using random initial solutions for all settings.

**Table 4** Averaged values of the standard deviation of measures calculated from 30 executions of CDPCA on 10 data sets randomly generated for each scenario with $J = 1000$ and $P = Q$

| Scenario | | bcd | | $\|\|E\|\|_2^2$ | | Var(%) | | ARI | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | | Objects | | Variables | |
| $(I, P)$ | $\epsilon$ | ALS | SDP | ALS | SDP | ALS | SDP | ALS | SDP | ALS | SDP |
| (10, 2) | 0.1 | 0.01 | 0.00 | 0.00 | 0.00 | 0.01 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| | 1 | 0.05 | 0.05 | 0.01 | 0.00 | 0.11 | 0.02 | 0.02 | 0.00 | 0.00 | 0.00 |
| (40, 2) | 0.1 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| | 1 | 0.44 | 0.03 | 0.01 | 0.00 | 0.04 | 0.01 | 0.03 | 0.00 | 0.00 | 0.00 |
| (40, 3) | 0.1 | 0.00 | 8.97 | 0.00 | 0.61 | 0.00 | 1.58 | 0.00 | 0.16 | 0.00 | 0.01 |
| | 1 | 0.11 | 6.87 | 0.02 | 0.13 | 0.34 | 0.58 | 0.06 | 0.18 | 0.05 | 0.02 |

### 3.3 Empirical analysis on real data sets

In order to compare the behaviour of the ALS and two-step-SDP algorithms in CDPCA modelling, we analyzed, in detail, the results of the application of both procedures on three real data sets.

#### 3.3.1 Data sets

The data sets used in the present study are characterized to have more attributes (genes) than objects (samples) and to have objects naturally clustered by $P = 2, 3, 4$ disjoint groups, namely:

- *leukemia*—available in the R package `plsgenomics` (Boulesteix et al. 2015); contains $J = 3051$ genes and $I = 38$ samples extracted from $P = 2$ types of tumor (dimension of each type-group: 11/27).
- *lymphoma*—available in the R package `spls` (Chung et al. 2013); contains $J = 4026$ genes and $I = 62$ samples extracted from $P = 3$ types of cancer (dimension of each type-group: 42/9/11).
- *SRBCT*—available in the R package `plsgenomics`; contains $J = 2308$ genes and $I = 83$ samples extracted from $P = 4$ different groups representing four small round blue cell tumor variants (dimension of each type-group: 29/11/18/25).

#### 3.3.2 Numerical results

We considered the fitting of each data set by CDPCA models with $Q = 2, 3, 4$ sparse and disjoint PCs. For each choice of $Q$, the parameters of the CDPCA model were estimated using both ALS and SDP approaches. Summarized results of several statistical measures used to compare the fitted models in our experiments are displayed on Tables 5, 6, 7, 8 and 9.

**Table 5** Numerical results using the ALS algorithm with different choices for $Q$

| Data set | $Q$ | Time (s) | Iter | bcd | $\|\|E\|\|_2^2$ | CDPCA components | |
|---|---|---|---|---|---|---|---|
| | | | | | | Non-sparsity (%) | Explained variance (%) |
| *Leukemia* ($P = 2$) | 2 | 3388.2 | 5 | 78.4 | 8.43 | 51.1/48.9 | 7.75/7.23 |
| | 3 | 2920.4 | 3 | 79.7 | 8.44 | 34.3/33.6/32.1 | 4.98/4.96/4.68 |
| | 4 | 2786.9 | 5 | 77.7 | 8.43 | 26.2/25.3/24.3/24.3 | 4.10/3.92/3.57/3.51 |
| *Lymphoma* ($P = 3$) | 2 | 43,614.5 | 12 | 87.7 | 7.12 | 51.3/48.7 | 12.82/12.50 |
| | 3 | 49,893.2 | 19 | 85.7 | 7.08 | 39.5/36.1/24.3 | 13.30/9.57/4.25 |
| | 4 | 66,769.9 | 15 | 85.0 | 7.05 | 31.3/29.7/19.0/20.0 | 10.30/10.23/3.91/3.55 |
| *SRBCT* ($P = 4$) | 2 | 12,786.9 | 15 | 90.4 | 4.90 | 56.4/43.6 | 8.75/6.37 |
| | 3 | 25,831.4 | 32 | 89.9 | 4.87 | 34.6/34.1/31.3 | 5.93/5.54/5.27 |
| | 4 | 28,566.7 | 12 | 81.3 | 4.92 | 31.2/24.6/22.9/21.3 | 5.89/4.86/4.40/4.25 |

**Table 6** Numerical results using the two-step-SDP algorithm with different choices for $Q$

| Data set | $Q$ | Time (s) | Iter | bcd | $\|\|E\|\|_2^2$ | CDPCA components | |
|---|---|---|---|---|---|---|---|
| | | | | | | Non-sparsity (%) | Explained variance (%) |
| *Leukemia*($P = 2$) | 2 | 60.78 | 2 | 75.9 | 8.43 | 55.0/45.0 | 7.83/7.63 |
| | 3 | 156.62 | 2 | 66.6 | 8.44 | 38.9/32.4/28.7 | 7.94/5.19/4.37 |
| | 4 | 157.99 | 12 | 61.4 | 8.43 | 29.4/27.6/24.9/18.1 | 6.92/6.28/2.98/2.95 |
| *Lymphoma*($P = 3$) | 2 | 373.08 | 3 | 92.8 | 7.22 | 53.4/46.6 | 10.90/10.72 |
| | 3 | 376.10 | 46 | 85.2 | 7.18 | 47.6/26.5/26.0 | 10.74/7.96/5.76 |
| | 4 | 425.40 | 134 | 81.1 | 7.15 | 28.4/26.9/23.3/21.5 | 7.49/6.70/6.35/6.13 |
| *SRBCT* ($P = 4$) | 2 | 28.97 | 2 | 86.2 | 4.99 | 56.0/44.0 | 6.28/5.63 |
| | 3 | 74.99 | 21 | 79.7 | 4.95 | 40.0/33.5/26.5 | 5.53/4.96/4.75 |
| | 4 | 70.59 | 3 | 77.3 | 4.92 | 34.5/27.8/21.3/16.5 | 5.36/4.93/4.62/2.06 |

**Table 7** Results for the $Q$ components constructed by the CDPCA (using both algorithms ALS and two-step-SDP), the standard PCA, and the robSPCA methodology

| Data set | $Q$ | Explained variance (%) | | | | robSPCA components |
|---|---|---|---|---|---|---|
| | | ALS | SDP | PCA | robSPCA | Non-sparsity (%) |
| *Leukemia*($J = 3051$) | 2 | 15.0 | 15.5 | 25.0 | 23.9 | 24.1/19.0 |
| | 3 | 14.6 | 17.5 | 34.9 | 30.2 | 20.4/15.2/11.8 |
| | 4 | 15.1 | 19.1 | 40.9 | 35.7 | 17.4/12.3/9.9/10.3 |
| *Lymphoma*($J = 4026$) | 2 | 25.3 | 21.6 | 29.9 | 28.3 | 22.3/14.7 |
| | 3 | 27.1 | 24.5 | 35.8 | 33.6 | 17.9/11.0/9.7 |
| | 4 | 28.0 | 26.7 | 41.0 | 38.3 | 15.4/9.3/7.7/5.8 |
| *SRBCT* ($J = 2308$) | 2 | 15.1 | 11.9 | 18.6 | 17.9 | 93.1/93.1 |
| | 3 | 17.0 | 15.2 | 26.5 | 25.4 | 93.3/93.1/93.2 |
| | 4 | 19.4 | 17.0 | 32.0 | 30.8 | 22.7/18.5/17.5/15.5 |

Similar patterns highlighted and reported previously in the simulation studies on the ALS and two-step-SDP are also observed on the real data sets under analysis.

Regarding the computation time, it is clearly observed that the SDP algorithm is faster (Tables 5, 6).

Evaluating the quality of the fitting, both algorithms show similar behaviour: in terms of the estimated model (norm) error, very similar values were produced by both algorithms with a tendency of the error to become lower for ALS ($0 \leq ||E||^2_{ALS} - ||E||^2_{SDP} \leq 0.1$).

Considering now the ability of the $Q$ CDPCA components to explain the variability of the data, the ALS algorithm exhibited better performance, in particular, on data sets with higher number $P$ of object clusters. In fact, the results show that, in general, the between cluster deviance (bcd), i.e., the total variance of the data in the reduced space, is higher for the ALS algorithm than for two-step-SDP, where

**Table 8** Clustering validation statistics for the CDPCA using the algorithms ALS and two-step-SDP, the robSPCA followed by k-means and the k-mean technique

| Measure | Data set | $Q$ | CDPCA | | k-mean on robSPCA | k-means |
|---|---|---|---|---|---|---|
| | | | ALS | SDP | Mean (cv) | Mean (cv) |
| ARI | *Leukemia* ($P = 2$) | 2 | 0.115 | 0.455 | 0.219 (0.929) | 0.543 (0.625) |
| | | 3 | 0.115 | 0.115 | 0.265 (1.165) | |
| | | 4 | 0.115 | 0.115 | 0.251 (0.893) | |
| | *Lymphoma* ($P = 3$) | 2 | 0.445 | 0.388 | 0.504 (0.266) | 0.394 (0.356) |
| | | 3 | 0.420 | 0.430 | 0.459 (0.250) | |
| | | 4 | 0.408 | 0.408 | 0.481 (0.386) | |
| | *SRBCT* ($P = 4$) | 2 | 0.102 | 0.032 | 0.077 (0.481) | 0.113 (0.742) |
| | | 3 | 0.100 | 0.026 | 0.076 (0.505) | |
| | | 4 | 0.069 | 0.042 | 0.057 (0.504) | |
| VI | *Leukemia* ($P = 2$) | 2 | 1.051 | 0.667 | 0.911 (0.230) | 0.539 (0.511) |
| | | 3 | 1.051 | 1.051 | 0.867 (0.305) | |
| | | 4 | 1.051 | 1.051 | 0.835 (0.225) | |
| | *Lymphoma* ($P = 3$) | 2 | 0.812 | 0.821 | 0.796 (0.189) | 0.810 (0.295) |
| | | 3 | 0.797 | 0.918 | 0.787 (0.184) | |
| | | 4 | 0.894 | 0.894 | 0.772 (0.319) | |
| | *SRBCT* ($P = 4$) | 2 | 2.114 | 2.409 | 2.202 (0.045) | 1.963 (0.161) |
| | | 3 | 2.114 | 2.442 | 2.386 (0.004) | |
| | | 4 | 2.254 | 2.394 | 2.323 (0.052) | |
| ASW | *Leukemia* ($P = 2$) | 2 | 0.688 | 0.686 | 0.375 (0.068) | 0.098 (0.124) |
| | | 3 | 0.684 | 0.562 | 0.344 (0.147) | |
| | | 4 | 0.071 | 0.502 | 0.270 (0.024) | |
| | *Lymphoma* ($P = 3$) | 2 | 0.609 | 0.584 | 0.506 (0.021) | 0.132 (0.065) |
| | | 3 | 0.626 | 0.521 | 0.385 (0.037) | |
| | | 4 | 0.616 | 0.494 | 0.333 (0.083) | |
| | *SRBCT*($P = 4$) | 2 | 0.667 | 0.447 | 0.438 (0.073) | 0.086 (0.265) |
| | | 3 | 0.670 | 0.531 | 0.441 (0.040) | |
| | | 4 | 0.554 | 0.545 | 0.368 (0.067) | |
| Dunn | *Leukemia* ($P = 2$) | 2 | 0.312 | 0.386 | 0.085 (0.213) | 0.645 (0.083) |
| | | 3 | 0.314 | 0.308 | 0.107 (0.283) | |
| | | 4 | 0.582 | 0.341 | 0.143 (0.173) | |
| | *Lymphoma* ($P = 3$) | 2 | 0.273 | 0.059 | 0.125 (0.112) | 0.646 (0.038) |
| | | 3 | 0.359 | 0.240 | 0.155 (0.062) | |
| | | 4 | 0.311 | 0.165 | 0.158 (0.056) | |
| | *SRBCT* ($P = 4$) | 2 | 0.339 | 0.021 | 0.112 (0.110) | 0.338 (0.040) |
| | | 3 | 0.431 | 0.251 | 0.169 (0.029) | |
| | | 4 | 0.216 | 0.288 | 0.118 (0.105) | |

the value of bcd decreases as the number $Q$ of components increases (Tables 5, 6). Analyzing the variability of the data in the original space, Tables 5 and 6 also show that the value of variance associated to the first CDPCA component constructed by the two-step-SDP algorithm is higher than that constructed using the ALS algorithm only for the *leukemia* data set (case $P = 2$). For the remaining data sets ($P = 3, 4$), the ALS algorithm presented the highest values. For all data sets, these highest values of variance decrease as the number $Q$ of components increases. Comparing the total variance of the $Q$ components, similar relationships are depicted in Table 7, namely, the two-step-SDP provides higher values for the case $P = 2$ (*leukemia*), while ALS yielded better results for $P = 3, 4$ (*lymphoma* and *SRBCT*).

Comparing with other dimensionality reduction techniques, both CDPCA (independently of the heuristic used) and robSPCA components present lower proportion of explained variance than that obtained by standard PCA (with normalized data), as expected, due to the existence of zero loadings into the first two cases (Table 7). Furthermore, the robSPCA method outperforms CDPCA providing higher proportion of explained variance even presenting higher sparsity levels (cf. the last two columns of Tables 5, 6 with Table 7). These differences of proportions become smaller as $Q$ increases. It is worthwhile mentioning that both R-functions provide the output of the $Q$ CDPCA components sorted in descending order of their variances.

Considering the proportion of original attributes included in each CDPCA component, i.e., its level of non-sparsity, both algorithm performs as expected: the lower the sparsity of a component is, the higher the percentage of variance explained is (see Tables 5, 6). Notice that, by definition of CDPCA, each original attribute (gene) has been included in only one CDPCA component. This does not hold for the robSPCA methodology. Results from Table 7 show that, for all the nine explored cases, the total level of non-sparsity of the robSPCA components either is lower than 100% (meaning that some genes are not included in any component) or is greater than 100% (meaning that there are genes included in more than one component), and in the latter case, it may be a drawback for interpretation purposes.

Comparing the quality of the clustering solution predicted by CDPCA model for each of three data sets, while both algorithms have exhibited low ability to recover the real object clustering on each three data sets, there is no consensus about which algorithm produces better estimates on the object clustering structure. The ALS algorithm, in general, produced better object clustering in terms of between cluster deviance and silhouette and Dunn indexes, suggesting the identification of more homogenous, compact and well-separated groups. In fact, results from Table 9 show that the ALS approach offered often better results in terms of the goodness of clustering structure, when measured by ASW and Dunn indexes, and in terms of the similarity of the estimated classification with the real cluster membership, when measured by ARI. Using VI index, the two-step-SDP algorithm was better, in particular, for the *SRBCT* data set ($P = 4$). Moreover, comparatively with the quality of the k-means clustering constructed in the original attributte space, both ALS and two-step-SDP algorithms for CDPCA presented a similar trend, namely, higher values of ARI for *lymphoma* ($P = 3$), and higher values of VI and ASW for the three data sets. Regarding the clustering obtained in the reduced space of the robSPCA components, although different

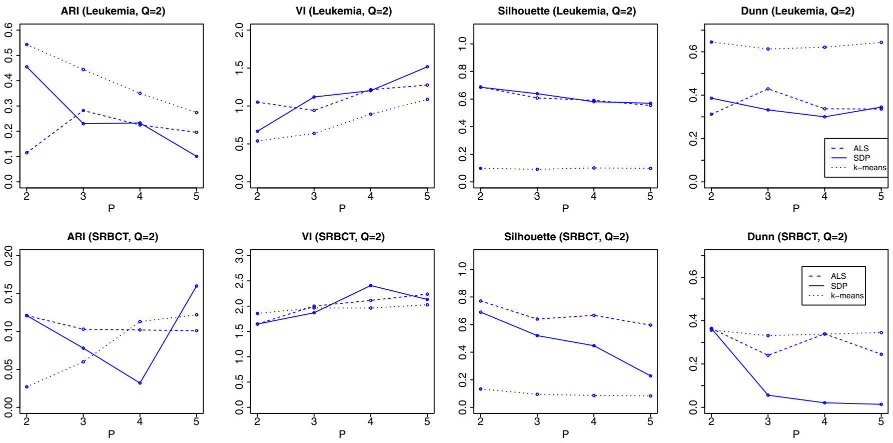

**Fig. 1** Measures of quality of the clustering for *leukemia* (on the top) and *SBCRT* (on the bottom) for $P = 2, 3, 4, 5$ object clusters and $Q = 2$ components given by CDPCA and given by the k-means technique

**Table 9** Values of the pseudo-F statistic $pF$ for *leukemia* data set

| | ALS | | | SDP | | |
|---|---|---|---|---|---|---|
| | $Q = 2$ | $Q = 3$ | $Q = 4$ | $Q = 2$ | $Q = 3$ | $Q = 4$ |
| $P = 2$ | 4.155 | 4.160 | 4.152 | 4.731 | 4.772 | 4.571 |
| $P = 3$ | 5.920 | 6.538 | 6.922 | 5.737 | 6.460 | 6.614 |
| $P = 4$ | 6.896 | 7.524 | 7.737 | 6.326 | 7.533 | 7.690 |

performances have been detected, in particular, for *SRBCT* ($P = 4$), where ALS yielded clusters with higher (lower, resp.) values of ARI (VI, resp.) in opposition to two-step-SDP, both algorithms revealed, in general, higher values of ASW and Dunn for the three data sets. Regardless of the data set, the variabilities of the four clustering validation measures were lower in the reduced space of the rob-SPCA components than in the the original attribute space, except for the Dunn index.

At the end, we analyzed whether there are differences in recovering the ‘true’ number $P$ of object clusters between the two existing algorithms for CDPCA. Since the number of object clusters is known for our data sets, we fix one parameter ($Q$) and consider the selection of $P$ as mentioned in Sect. 2.4. In particular, based on the four clustering validation measures and fixing $Q = 2$, the problem of determining $P$ for the *leukemia* and *SRBCT* data sets is here presented (Fig. 1). For the first data set, two-step-SDP and the k-means technique tended to get the correct answer ($P = 2$) using ARI, ASW and Dunn indexes, but, for *SRBCT* data, the number of object clusters was only correctly detected ($P = 4$) when the VI index on the two-step-SDP algorithm was examined. Therefore, it was not possible to identify a single measure that leads to the correct cluster solution on different data sets. Nevertheless, CDPCA involves the choice of both the number of object clusters and variable partition. Thus, based on the *pF* index, a choice of the pair ($P$, $Q$) might be suggested. For instance, for the *leukemia* data set, varying $P$ and $Q$ between 2 and 4, both algorithms agree that the pair ($P$, $Q$), corresponding to the highest value of the statistic pseudo-$F$, will be taking $P = Q = 4$ (Table 9) but in fact $P = 2$. Restricting to the case $P = 2$, the value of *pF* indicates $Q = 3$ as the suitable number for variable partition for both algorithms. Fixing $Q = 2$ as above, *pF* suggests $Q = 4$ as the suitable number of components. This lack of ability of all those measures in providing the correct answer, in particular, for the ALS heuristic, shows that further investigation should focus on a new statistics that allows to select the suitable number of $P$ and $Q$ for practical applications of the CDPCA method.

## 4 Conclusions

CDPCA is a two-mode methodology where the data is described by disjoint components with the objects classified by clusters. The disjoint component loadings make easier the interpretation of the components when compared to the standard PCA, and induce a proportion of sparsity equal to $(Q-1)/Q$ $(\geq 0.5, \forall Q \geq 2)$ in the $(J \times Q)$ component loading matrix. In this sense, CDPCA can be considered a sparse PCA with the advantage that it does not depend on a prior knowledge on the level of sparsity of its components.

In this paper, the behaviour of two heuristic procedures proposed in the literature to estimate the parameters of the CDPCA models when fitted to high-dimensional data is explored. We started by briefly describing those procedures for performing CDPCA on two-way data. Then, we proceeded with a comparative analysis of the results provided by the two algorithms side-by-side and on simulated and real data sets. Three real data sets with different number of clusters of objects were chosen. We chose different number of components and discussed the results on applying the `R` functions for CDPCA: `CDpca` and `TwostepSDPClust`.

Although the number of clusters and components, and the diversity of settings concerning the sample size and the number of variables considered in this study have been limited due to the computational effort involving with the ALS algorithm, our findings show interesting patterns. From our computational tests we can conclude:

- Both algorithms are not affected by randomly generated initial solutions, whenever a sufficient number of runs (`r`) in the ALS algorithm is considered. From our experience, it is recommended to run at least 10 times (`r` $\geq$ 10) for high dimensional data sets;
- Regardless the number of object clusters and the number of sparse and disjoint components provided by the estimated model, the two-step-SDP algorithm (using the `R` function `TwostepSDPClust`) shows a significant improvement in terms of computational time when compared with the ALS algorithm (using the `R`-function `CDpca`);

- The ALS algorithm tends to provide slightly better estimated models in terms of solution precision, when measured using the Frobenius norm of the error in the CDPCA model;
- The algorithm two-step-SDP outperforms ALS in terms of proportion of variance explained by the disjoint components when $P = 2$, while in other cases they yield similar values;
- While both algorithms present good performance to recover the true object clusters, the two-step-SDP is better for lower sample size and higher error level in the CDPCA model ($\epsilon = 2$);
- In general, the complexity of the original data structure tends to decrease the proportion of variance explained by the disjoint components induced by the correlation structure of the data and the ability of these components to recover the true variable partition;
- In general, ALS procedure provides the identification of more homogenous, compact and well-separated groups of objects;
- Although there was no single quality measure consensual for the three data sets, results suggest the two-step-SDP algorithm presents more ability to recover the true number of object clusters;
- Concerning the partitioning of the variables, the two algorithms have different purposes, and then may produce quite different estimates in CDPCA modelling;
- The two-step-SDP algorithm performs better in recovering the true variable partition;
- The ALS algorithm may be less suitable to perceiving the clustering structure of the variables, in particular, when there are two object clusters in the data set ($P = 2$).

In conclusion, based on the presented experiments, the two-step-SDP approach seems to be a major tool in terms of getting results faster and with a great ability to recover the true number of object clusters, while the ALS algorithm outperforms by providing more accurate results in the reduced space of the components, identifying more clearly homogeneous, compact and well-separated clusters.

**Acknowledgements** The authors are grateful to the reviewers for their comments and suggestions that helped to greatly improve the quality of this paper. Special thanks to Giorgia Zaccaria for her valuable comments on the pseudo-F statistic. This work is supported by the Center for Research and Development in Mathematics and Applications (CIDMA) through the Portuguese Foundation for Science and Technology (FCT—Fundação para a Ciência e a Tecnologia) references UIDB/04106/2020 and UIDP/04106/2020, and by the projects UIDB/00481/2020 and UIDP/00481/2020—FCT—and CENTRO-01-0145-FEDER-022083—Centro Portugal Regional Operational Programme (Centro2020), under the PORTUGAL 2020 Partnership Agreement, through the European Regional Development Fund.

## References

Adachi K, Trendafilov NT (2016) Sparse principal component analysis subject to prespecified cardinality of loadings. Comput Stat 31(4):1403–1427

Boulesteix AL, Durif G, Lambert-Lacroix S, Peyre J, Strimmer K (2015) plsgenomics: PLS Analyses for Genomics, R package version 1.3-1 https://CRAN.R-project.org/package=plsgenomics

Calinski T, Harabasz J (1974) A dendrite method for cluster analysis. Commun Stat 3:1–27

Cavicchia C, Vichi M, Zaccaria G (2020) The ultrametric correlation matrix for modelling hierarchical latent concepts, Adv Data Anal Classif. https://doi.org/10.1007/s11634-020-00400-z

Charrad M, Ghazzali N, Boiteau V, Niknafs A (2014) NbClust: an R package for determining the relevant number of clusters in a data set. J Stat Softw 61(6):1–36

Chung D, Chun H, Keles S (2013) spls: sparse partial least squares (SPLS) regression and classification. R package version 2.2-1. https://CRAN.R-project.org/package=spls

d'Aspremont A, El Ghaoui L, Jordan MI, Lanckriet GRG (2007) A direct formulation for sparse PCA using semidefinite programming. SIAM 49(3):434–448

DeSarbo WS, Jedidi K, Cool K, Schendel D (1990) Simultaneous multidimensional unfolding and cluster analysis: an investigationof strategic groups. Mark Lett 2:129–146

Enki DG, Trendafilov NT, Jolliffe IT (2013) A clustering approach to interpretable principal components. J Appl Stat 40(3):583–599

Erichson NB, Zheng P, Aravkin S (2018) sparsepca: Sparse Principal Component Analysis (SPCA), R package version 0.1.2. https://CRAN.R-project.org/package=sparsepca

Erichson NB, Zheng P, Manohar K, Brunton S, Kutz JN, Aravkin AY (2018) Sparse principal component analysis via variable projection. IEEE J Sel Top Signal Process (available at arXiv 1804.00341)

Hennig C (2015) fpc: Flexible Procedures for Clustering. R package version 2.1-10. https://CRAN.R-project.org/package=fpc

Hunter MA, Takane Y (2002) Constrained principal component analysis: various applications. J Educ Behav Stat 27:41–81

Jolliffe IT (2002) Principal component analysis, 2nd edn. Springer, New York

Jolliffe IT, Trendafilov NT, Uddin M (2003) A modified principal component technique based on the lasso. J Comput Graph Stat 12(3):531–547

Ma Z (2013) Sparse principal component analysis and iterative thresholding. Ann Stat 41(2):772–801

Macedo E (2015) Two-step-SDP approach to clustering and dimensionality reduction. Stat Optim Inf Comput 3(3):294–311

Macedo E, Freitas A (2015) The alternating least-squares algorithm for CDPCA. In: Plakhov A et al (eds) Optimization in the natural sciences, communications in computer and information science (CCIS), vol 499. Springer, pp 173–191

Nieto-Librero AB, Galindo-Villardón MP, Freitas A (2019)biplotbootGUI: Bootstrap on Classical Biplots and Clustering Disjoint Biplot, R package version 1.2. http://www.R-project.org/package=biplotbootGUI

Nieto-Librero AB, Sierra C, Vicente-Galindo MP, Ruíz-Barzola O, Galindo-Villardón MP (2017) Clustering disjoint HJ-Biplot: a new tool for identifying pollution patterns in geochemical studies. Chemosphere 176:389–396

Overton ML, Womersley RS (1993) Optimality conditions and duality theory for minimizing sums of the largest eigenvalues of symmetric matrices. Math Program 62:321–357

Peng J, Wei Y (2007) Approximating k-means-type clustering via semidefinite programming. SIAM J Optim 18(1):186–205

Peng J, Xia Y (2005) A new theoretical framework for k-means-type clustering. In: Chu W et al (eds) Foundations and advances in data mining studies in fuzziness and soft computing, vol 180. Springer, pp 79–96

R Development Core Team (2019) R: a language and environment for statistical computing. http://www.R-project.org/

Rocci R, Vichi M (2008) Two-mode multi-partitioning. Comput Stat Data Anal 52:1984–2003

Takane Y, Hunter MA (2001) Constrained principal component analysis: a comprehensive theory. Appl Algebra Eng Commun Comput 12:391–419

Vichi M (2017) Disjoint factor analysis with cross-loadings. Adv Data Anal Classif 11(3):563–591

Vichi M, Saporta G (2009) Clustering and disjoint principal component analysis. Comput Stat Data Anal 53:3194–3208

Vines S (2000) Simple principal components. Appl Stat 49:441–451

Xu R, Wunsch D (2005) Survey of clustering algorithms. IEEE Trans Neural Netw 16:645–648

Zou H, Hastie T, Tibshirani R (2006) Sparse principal component analysis. J Comput Graph Stat 15(2):262–286

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

## Affiliations

**Adelaide Freitas**[1,2] · **Eloísa Macedo**[3] · **Maurizio Vichi**[4]

✉ Adelaide Freitas
adelaide@ua.pt

[1] Department of Mathematics, University of Aveiro, 3810-193 Aveiro, Portugal

[2] CIDMA—Center for Research and Development in Mathematics and Applications, University of Aveiro, 3810-193 Aveiro, Portugal

[3] TEMA—Center for Mechanical Technology and Automation, University of Aveiro, 3810-193 Aveiro, Portugal

[4] Department of Statistical Sciences, University "La Sapienza", P.le A. Moro 5, 00185 Rome, Italy