

The alternating least-squares algorithm for CDPCA *

Eloísa Macedo, Department of Mathematics & CIDMA,
University of Aveiro, Portugal (macedo@ua.pt)
Adelaide Freitas, Department of Mathematics & CIDMA,
University of Aveiro, Portugal (adelaide@ua.pt)

Abstract

Clustering and Disjoint Principal Component Analysis (CDPCA) is a constrained principal component analysis recently proposed for clustering of objects and partitioning of variables, simultaneously, which we have implemented in R language. In this paper, we deal in detail with the alternating least-squares algorithm for CDPCA and highlight its algebraic features for constructing both interpretable principal components and clusters of objects. Two applications are given to illustrate the capabilities of this new methodology.

Key Words and Phrases: principal component analysis, clustering, k-means.

1 Introduction

Principal Component Analysis (PCA) is a widely used tool in applied statistics for exploratory data analysis and dimensionality reduction. It has many important applications in different fields, such as neuroscience, computer graphics, image compression, meteorology, oceanography, and in gene expression [2].

In essence, PCA allows the reduction of the dimensionality of data by the detection of a lower number of uncorrelated variables, called components, that are able to explain the maximum variability of the data, i.e., the data compression is done with minimum information loss. An orthogonal transformation projects the data into a lower dimensional space along the directions where the data presents the highest variability. This statistical technique is useful to represent data by drawing a low-dimensional graph (e.g., in biplots)

*Accepted authors manuscript (AAM) published by Springer in [EmC-ONS 2014, CCIS 499, pp. 173–191, 2015]. [DOI: MAL10.1007/978-3-319-12577-0-61]. The final publication is available at [link.springer.com](http://link.springer.com/chapter/10.1007%2F978-3-319-20352-2_12) via http://link.springer.com/chapter/10.1007%2F978-3-319-20352-2_12

in order to find patterns hidden on data and to interpret relationships between samples and variables. PCA can be performed via singular value decomposition of the data matrix.

Since each principal component (PC) is a linear combination of all the original variables, i.e., with nonzero loadings, this can be considered a tremendous shortcoming for component interpretation. To overcome this difficulty, various PCA-based methodologies have been proposed in the recent years, for instance, based on rotation techniques or obtaining components with zero loadings. In this latter context, several major papers have been published. In [9], it is proposed a new methodology called Simple Principal Component Analysis, which idea is to restrict the components' loadings to be equal to -1, 0 or 1. In 2003, Jolliffe, Trendafilov and Uddin [3] introduced SCoTLASS, which is a maximal variance approach that obtains components where a bound is introduced on the sum of the absolute values of the loadings, and some become zero. Later, in 2006, Zou, Hastie and Tibshirani [11] introduced the Sparse Principal Component Analysis, which aims to obtain modified principal components with sparse loadings. In [11], it is also proposed efficient algorithms to perform the new sparse PCA and some numerical experiments with real and simulated data are reported. In 2007, a new approach for sparse PCA via Semidefinite Programming was proposed in [1], based on a convex semidefinite relaxation of the sparse PCA problem. There are also reported numerical experiments for comparing that technique with others. More recently, in 2013, it is proposed in [4] a new sparse PCA and an iterative thresholding algorithm to estimate principal subspaces.

When dealing with real data sets, there may be the need of reducing not only the dimension of the variable space, but also to reveal some patterns among the objects. Obviously, this can be done by performing PCA on the variables and applying a clustering technique on the objects. The desirable scenario for data visualization and interpretation is to obtain non overlapping clusters of objects and disjoint or sparse principal components.

A new methodology called Clustering and Disjoint Principal Component Analysis (referred to hereafter as CDPCA) [8] was recently proposed for clustering of objects and partitioning of variables, simultaneously. It permits to cluster objects along a set of centroids and partition the variables into a reduced set of components, simultaneously, in order to maximize the between cluster deviance of the components in the reduced space. The CDPCA classification of data consists of the construction of groups based on the closeness and similarity among data.

In [8], the proposed CDPCA model is described as a joint model of K-means applied on the data matrix and PCA applied on the matrix of centroids. Hence, it depends on three parameter matrices: one matrix for allocating the objects into the clusters, one other for identifying the centroids and another one for identifying to the loading components. The least-squares estimators of these parameters can be obtained by solving a quadratic mixed continuous and integer optimization problem [8]. An alternating least-squares (ALS) algorithm based on four steps is suggested in [8] to solve the problem. Notice that the ALS algorithm can be considered as an heuristic that iteratively solves the optimization problem based on two basic steps: allocation of objects via K-means ([10]) and reduction of the variable space via application of PCA on the resulting centroids. In this paper,

we describe a detailed two-step-based scheme of the ALS algorithm proposed in [8] for estimating the parameters of the CDPCA model. Unlike PCA, in CDPCA disjoint components are returned, and thus, each original variable contributes to a single component. It is worth mentioning that the obtained CDPCA score components may be correlated, unlike in PCA where uncorrelated components are provided.

Recently, we have implemented the CDPCA in a easy-to-use software application [5] using R language [6], which is available from the authors upon request. Beside returning an assignment matrix for the allocation of objects into clusters and a component loading matrix which allows to allocate the variables into disjoint subsets, the main features of our R-based implementation of CDPCA include a plot of the data projected into the two dimensional space defined by the first two CDPCA components, and also a pseudo-confusion matrix when the real classification is known, permitting to summarize and visualize the (mis)classification of the objects. The goal of this paper is to explain and illustrate the algebraic features of the two essential steps in each iteration of the ALS algorithm. A toy example is included to show some transformations performed in each step of the ALS algorithm. Additionally, a numerical experiment using real data is presented. To execute these analyses we use our R-implemented function of CDPCA. Since the goal of this work is not focused on our R-based implementation, only brief reference to this function will be given in the numerical example.

The paper is organized as follows. Section 2 presents the theoretical background and tools needed for the CDPCA technique. Section 3 is devoted to highlight the algebraic features behind the CDPCA detailing the ALS algorithm step by step. In Section 4, application of CDPCA using data from a breast cancer study is presented and the results are compared with those obtained using PCA. Concluding remarks appear in Section 5.

2 The methodology of CDPCA

In this section we describe the CDPCA, based on the paper [8].

2.1 Notation

First of all, let us introduce some notations and basic definitions that will be used throughout this work.

$\mathbf{X} = (x_{ij})$: Data matrix with I objects in rows and J variables in columns; \mathbf{X} is assumed to be standardized.

P, Q : Desired number of clusters of objects and subsets of variables, respectively.

$\mathbf{U} = (u_{ip})$: Matrix defining an allocation of the I objects into P clusters; \mathbf{U} is a $I \times P$

binary and row stochastic matrix defined as

$$\begin{cases} u_{ip} = 1, & \text{if the } i\text{-th object belongs to the cluster } p, \\ u_{ip} = 0, & \text{otherwise.} \end{cases}$$

$\mathbf{V} = (v_{jq})$: Matrix defining a partition of the J variables into Q subsets; \mathbf{V} is a $J \times Q$ binary and row stochastic matrix defined as

$$\begin{cases} v_{jq} = 1, & \text{if the } j\text{-th variable belongs to the subset } q, \\ v_{jq} = 0, & \text{otherwise.} \end{cases}$$

$\bar{\mathbf{X}}$: Object centroid matrix in the original space; $\bar{\mathbf{X}}$ is a $P \times J$ matrix defined by $\bar{\mathbf{X}} = (\mathbf{U}^T \mathbf{U})^{-1} \mathbf{U}^T \mathbf{X}$.

$\mathbf{Z} = (z_{ij})$: Centroid-based data matrix where each object is identified by the corresponding centroid, i.e., each object is projected into the space defined by the P clusters; \mathbf{Z} is a $I \times J$ matrix given by $\mathbf{Z} = \mathbf{U} \bar{\mathbf{X}}$.

$\mathbf{W}^{(q)} = (w_{ik}^{(q)})$: Submatrix extracted from the centroid-based data matrix \mathbf{Z} where only the original variables assigned into the q -th column of \mathbf{V} are considered; $\mathbf{W}^{(q)}$ is a $I \times K^{(q)}$ matrix defined as

$$w_{ik}^{(q)} = z_{ij}, \text{ if } v_{jq} = 1, \text{ with } k = \text{rank}_{J^{(q)}}(j),$$

where $J^{(q)} = \{j : v_{jq} = 1\}$, $K^{(q)} = \#J^{(q)}$ and $k = 1, \dots, K^{(q)}$.

$\mathbf{A} = (a_{jq})$: Matrix of the component loadings; \mathbf{A} is a $J \times Q$ matrix where the Q columns are identifying the coefficients of Q linear combinations (i.e., the Q principal components for CDPCA) such that $\text{rank}(\mathbf{A}) = Q$, $\mathbf{A}^T \mathbf{A} = \mathbf{I}_Q$ and $\sum_{j=1}^J (a_{jq} a_{jr})^2 = 0$, for any q and r ($q \neq r$).

$\mathbf{Y} = (y_{iq})$: Component score matrix where y_{iq} is the value of the i -th object for the q -th CDPCA component; \mathbf{Y} is a $I \times Q$ matrix given by $\mathbf{Y} = \mathbf{X} \mathbf{A}$.

$\bar{\mathbf{Y}}$: Object centroid matrix in the reduced space; $\bar{\mathbf{Y}}$ is a $P \times Q$ matrix defined by $\bar{\mathbf{Y}} = \bar{\mathbf{X}} \mathbf{A}$.

2.2 Model

The CDPCA model results from the application of PCA on the transformed data matrix, where each object is replaced by its centroid. By its turn, the centroids are obtained by applying the K-means algorithm on the original data matrix ([8]).

Hence, the data matrix would be fitted by the model

$$\begin{aligned}
\mathbf{X} &= \mathbf{U}\bar{\mathbf{X}} + \mathbf{E}_1 && \text{(K-means applied on } \mathbf{X}\text{)} \\
&= \mathbf{U}\bar{\mathbf{Y}}\mathbf{A}^T + \mathbf{E}_1 + \mathbf{E}_2 && \text{(PCA applied on } \mathbf{U}\bar{\mathbf{X}}\text{)} \\
&= \mathbf{U}\bar{\mathbf{Y}}\mathbf{A}^T + \mathbf{E} && \text{(CDPCA model)}
\end{aligned} \tag{1}$$

where \mathbf{E} , \mathbf{E}_1 , \mathbf{E}_2 are $I \times J$ error matrices with $\mathbf{E} = \mathbf{E}_1 + \mathbf{E}_2$.

2.3 Optimization problem

From the CDPCA model (1), it is easy to see that $\mathbf{E} = \mathbf{X} - \mathbf{U}\bar{\mathbf{Y}}\mathbf{A}^T$. Therefore, the CDPCA problem intends to minimize the norm of the error matrix \mathbf{E} , resulting in the following optimization problem

$$\min_{\mathbf{U}, \bar{\mathbf{Y}}, \mathbf{A}} \|\mathbf{X} - \mathbf{U}\bar{\mathbf{Y}}\mathbf{A}^T\|^2, \tag{2}$$

subject to the above conditions for the matrices \mathbf{U} (i.e., \mathbf{U} is a binary and row stochastic matrix), $\bar{\mathbf{Y}}$ (i.e., $\bar{\mathbf{Y}}$ is an object centroid matrix in the reduced space) and \mathbf{A} (i.e., \mathbf{A} is a columnwise orthonormal matrix where each row contributes to a single column).

It can be proved that the problem (2) is equivalent to the maximization of the between cluster deviance $\|\mathbf{U}\bar{\mathbf{X}}\mathbf{A}\|^2$ of the components in the reduced space, subject to constraints on the matrices \mathbf{U} and \mathbf{A} . Since the decomposition $\|\mathbf{X} - \mathbf{U}\bar{\mathbf{Y}}\mathbf{A}^T\|^2 = \|\mathbf{X}\|^2 - \|\mathbf{U}\bar{\mathbf{Y}}\mathbf{A}^T\|^2$ holds ([8]), the above problem (2) is equivalent to

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

subject to the same constraints of problem (2). Since $\bar{\mathbf{Y}} = \bar{\mathbf{X}}\mathbf{A}$ and \mathbf{A} has orthonormal columns (i.e., $\mathbf{A}^T\mathbf{A} = \mathbf{I}$), then $\|\mathbf{U}\bar{\mathbf{Y}}\mathbf{A}^T\|^2 = \|\mathbf{U}\bar{\mathbf{X}}\mathbf{A}\|^2$. Hence, problem (3) is equivalent to

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

To solve this optimization problem, the authors of CDPCA proposed the inclusion of the matrix \mathbf{V} described in Section 2.1 which specifies the partition of J variables into Q disjoint components. The positions of the nonzero elements of the matrix \mathbf{A} are identified by the positions of the one's in the matrix \mathbf{V} . Hence, and since $\bar{\mathbf{Y}} = \bar{\mathbf{X}}\mathbf{A}$, the CDPCA problem can be formulated as the following quadratic mixed continuous and integer

problem:

$$\begin{aligned}
\max \quad & F = \|\mathbf{U}\bar{\mathbf{Y}}\|^2 \\
\text{s. t.} \quad & u_{ip} \in \{0, 1\}, \quad i = 1, \dots, I; \quad p = 1, \dots, P \\
& \sum_{p=1}^P u_{ip} = 1, \quad i = 1, \dots, I \\
& v_{jq} \in \{0, 1\}, \quad j = 1, \dots, J; \quad q = 1, \dots, Q \\
& \sum_{q=1}^Q v_{jq} = 1, \quad j = 1, \dots, J \\
& \sum_{j=1}^J a_{jq}^2 = 1, \quad q = 1, \dots, Q \\
& \sum_{j=1}^J a_{jq}a_{jr} = 0, \quad q = 1, \dots, Q-1; \quad r = q+1, \dots, Q
\end{aligned} \tag{5}$$

The first two constraints in (5) correspond to the allocation of I objects into P clusters. The following two constraints represent the allocation of J variables into Q disjoint subsets of variables (components). The remaining constraints are associated to the PCA implementation. The objective function value is calculated by $\|\mathbf{U}\bar{\mathbf{Y}}\|^2 = \text{tr}(\mathbf{U}\bar{\mathbf{Y}}(\mathbf{U}\bar{\mathbf{Y}})^T)$, corresponding to the between cluster distances. Based on linear algebra properties, the objective function value F can also be equivalently computed by $\text{tr}((\mathbf{U}\bar{\mathbf{Y}})^T\mathbf{U}\bar{\mathbf{Y}})$, representing the total variance of the data in the reduced space, where the objects are identified by their centroids. The main goal is the achievement of maximum dissimilarity or distance between centroids (and objects) of different clusters. The idea of CDPCA is finding a clustering of objects along a set of centroids and, simultaneously, a partition of variables along a reduced set of disjoint components, in order to maximize the between cluster deviance in the reduced space of the disjoint components.

2.4 Algorithm

In [8], it is proposed an iterative algorithm called alternating least-squares algorithm (ALS) to solve the optimization problem (5). Each iteration of the ALS algorithm can be summarily described by two basic steps: allocation of objects via K-means and reduction of the variable space via application of PCA on the resulting centroids. Concretely,

- Step 1: Concerning to the objects:

allocate the I objects into P clusters (matrix \mathbf{U}),
calculate the centroids in the space of the observed variables (matrix $\bar{\mathbf{X}}$)
identify the objects by its cluster centroids in the space of the observed variables (matrix \mathbf{Z}).

- Step 2: Concerning to the variables:

allocate the J variables into Q subsets (matrix \mathbf{V}),

obtain the loadings of the CDPCA components (matrix \mathbf{A}),
 calculate the centroids in the reduced space of the Q CDPCA components
 (matrix $\bar{\mathbf{Y}}$),
 identify the objects in the reduced space of the Q CDPCA components (matrix
 \mathbf{Y}).

These steps are summarized in Figure 1. At the beginning, in Step 1 and with the standardized data matrix \mathbf{X} of I objects described by J variables, the I objects are assigned into P clusters by means of the matrix \mathbf{U} . Next, each row of the data matrix is replaced by its corresponding object centroid resulting then in the matrix \mathbf{Z} . In Step 2, the allocation of the J variables into Q disjoint subsets is specified in the matrix \mathbf{V} and the CDPCA component loadings are specified in the matrix \mathbf{A} . To obtain these two matrices, an iterative process working row-by-row and column-by-column of the matrices \mathbf{V} and \mathbf{A} is executed in order to maximize the objective function F . At the end of Step 2, the component score matrix, \mathbf{Y} , and the object centroid matrix in the reduced space, $\bar{\mathbf{Y}}$, are found as well as the value of the objective function F . Thus, at the end of one iteration of the algorithm, the I objects of the data matrix are allocated into P clusters, and simultaneously displayed in a reduced space of Q disjoint components. The value of the between cluster deviance is also calculated to evaluate the quality of the clustering of the I objects in the reduced space. In the next iteration, the process is repeated using \mathbf{Y} as the input data matrix. The iterative procedure of the algorithm stops when there is a difference between consecutive computations of the values of the objective function F smaller than a specified tolerance.

Since the function F is bounded above, the algorithm converges to a stationary point, which is at least a local maximum of problem ([8]). This procedure can be considered as an heuristic and thus, to guarantee that the global maximum is achieved, it has been suggested to run the algorithm several times for different initial allocation matrices \mathbf{U} and \mathbf{V} , which are randomly chosen at the beginning of each run.

3 CDPCA: step by step

In this section, we present in detail the main algebraic features of the ALS algorithm for performing CDPCA.

To show the main algebraic features of the CDPCA procedure, we have performed CDPCA on a synthetic data matrix \mathbf{X} constructed for satisfying the model (1) and where the objects are partitioned along a set of three clusters and the variables along a set of two components. For that purpose, we consider $I = 15$, $J = 3$, $P = 3$, $Q = 2$, and the

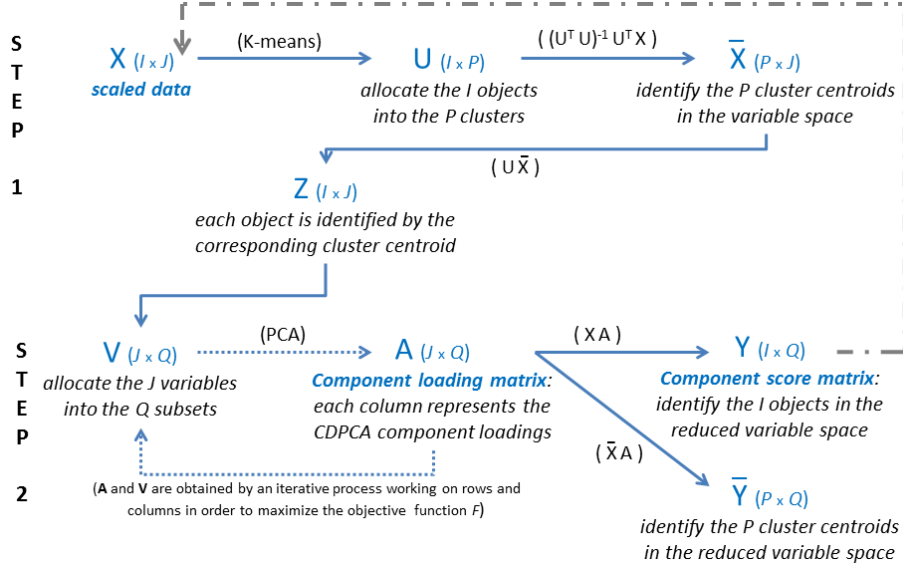


Figure 1: The two basic steps of one iteration of the ALS algorithm for performing CD-PCA.

following matrices satisfying the conditions mentioned in Section 2.1:

$$\mathbf{U} = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{bmatrix}, \quad \bar{\mathbf{Y}} = \begin{bmatrix} \sqrt{2/3} & -2 \\ \sqrt{2/3} & 1 \\ -\sqrt{3/2} & 1 \end{bmatrix} \quad \text{and} \quad \mathbf{A} = \begin{bmatrix} 1 & 0 \\ 0 & \sqrt{2}/2 \\ 0 & \sqrt{2}/2 \end{bmatrix}. \quad (6)$$

It is easy to check that, under these circumstances, $\mathbf{U}\bar{\mathbf{Y}}\mathbf{A}^T$ is a standardized matrix. An error \mathbf{E} is added to obtain the model (1). Herein we considered the matrix \mathbf{E} with values randomly generated of a normal distribution with mean zero and standard deviation equal to 0.8. Thus, we have

$$\mathbf{X} = \mathbf{U}\bar{\mathbf{Y}}\mathbf{A}^T + \mathbf{E}$$

being

$$\mathbf{U}\bar{\mathbf{Y}}\mathbf{A}^T = \begin{array}{c|cc} \begin{array}{c} 0.816 \\ 0.816 \\ 0.816 \\ 0.816 \\ 0.816 \end{array} & \begin{array}{c} -1.414 \\ -1.414 \\ -1.414 \\ -1.414 \\ -1.414 \end{array} & \begin{array}{c} -1.414 \\ -1.414 \\ -1.414 \\ -1.414 \\ -1.414 \end{array} \\ \hline \begin{array}{c} 0.816 \\ 0.816 \\ 0.816 \\ 0.816 \end{array} & \begin{array}{c} 0.707 \\ 0.707 \\ 0.707 \\ 0.707 \end{array} & \begin{array}{c} 0.707 \\ 0.707 \\ 0.707 \\ 0.707 \end{array} \\ \hline \begin{array}{c} -1.224 \\ -1.224 \\ -1.224 \\ -1.224 \\ -1.224 \\ -1.224 \end{array} & \begin{array}{c} 0.707 \\ 0.707 \\ 0.707 \\ 0.707 \\ 0.707 \\ 0.707 \end{array} & \begin{array}{c} 0.707 \\ 0.707 \\ 0.707 \\ 0.707 \\ 0.707 \\ 0.707 \end{array} \end{array} \quad \mathbf{X} = \begin{array}{c|cc} \begin{array}{c} 0.917 \\ 1.860 \\ 0.460 \\ 1.290 \\ 1.567 \end{array} & \begin{array}{c} -1.093 \\ -1.767 \\ -1.509 \\ -0.412 \\ -1.812 \end{array} & \begin{array}{c} -2.382 \\ -0.289 \\ -0.132 \\ -1.405 \\ -1.530 \end{array} \\ \hline \begin{array}{c} 0.982 \\ 0.832 \\ 2.461 \\ 0.697 \end{array} & \begin{array}{c} -0.167 \\ 1.710 \\ 1.315 \\ 1.520 \end{array} & \begin{array}{c} 1.531 \\ 0.334 \\ 1.203 \\ 1.519 \end{array} \\ \hline \begin{array}{c} -2.273 \\ -1.603 \\ -1.003 \\ -0.799 \\ -1.133 \\ -2.599 \end{array} & \begin{array}{c} 0.152 \\ 1.483 \\ -0.043 \\ 1.763 \\ 0.002 \\ 0.473 \end{array} & \begin{array}{c} 0.464 \\ -0.476 \\ -0.840 \\ -0.770 \\ 1.145 \\ 0.393 \end{array} \end{array}$$

The horizontal and vertical lines separate the three clusters of objects and the set of variables, respectively, and in accordance with (6).

Using the synthetic data matrix \mathbf{X} , we now focus on the algebraic features behind the two basic steps of the CDPCA methodology and afterwards illustrate some outputs obtained by our R-based application.

3.1 Initialization

Set $k = 0$. At the beginning, the data matrix \mathbf{X} is standardized:

$$x_{ij} \mapsto \frac{x_{ij} - \bar{x}_j}{\sqrt{\sum_{i=1}^I (x_{ij} - \bar{x}_j)^2 / I}}$$

where $\bar{x}_j = \sum_{i=1}^I x_{ij} / I$. Next, the parameters of the ALS algorithm to perform CDPCA are initialized as follows:

Step 1. Parameters associated to the objects:

- The matrix \mathbf{U}_0 is randomly generated such that there is only a nonzero element per row and that element is equal to 1 (i.e., \mathbf{U}_0 is the initial object assignment matrix).
- The object centroid matrix $\bar{\mathbf{X}}_0$ is computed. For such, the mean of each variable into each object cluster is calculated.

- All the objects are identified by its cluster centroids. This information is provided by the centroid-based data matrix \mathbf{Z} .

Step 2. Parameters associated to the components:

- The matrix \mathbf{V}_0 is randomly generated such that there is only a nonzero element per row and that element is equal to 1 (i.e., \mathbf{V}_0 is the initial variable assignment matrix)
- The CDPCA component loading matrix \mathbf{A}_0 is constructed column-by-column solving Q independent PCA subproblems, one by each column. The nonzero elements of the q -th column of \mathbf{V}_0 identify the original variables belonging to the q -th CDPCA component. These elements will be considered in the PCA subproblem to obtain the nonzero elements of the q -th column of \mathbf{A}_0 . Thus, the nonzero elements on the q -th column of \mathbf{A}_0 correspond to the first principal component obtained from PCA applied on the submatrix $\mathbf{W}_0^{(q)}$ which is extracted from the centroid-based data matrix $\mathbf{Z}_0 = \mathbf{U}_0 \bar{\mathbf{X}}_0$ (i.e., the data matrix where each object is identified by the corresponding centroid) and restricted to the original variables assigned into the q -th column of \mathbf{V}_0 . Therefore, the q -th column of \mathbf{A}_0 provides the direction vector with maximum variability among the centroids in the subspace defined by the original variables assigned to the q -th column of \mathbf{V}_0 .

3.2 General iteration

At the beginning of the $(k+1)$ -th iteration of the algorithm, the matrices \mathbf{U}_k , $\bar{\mathbf{X}}_k$, \mathbf{V}_k , \mathbf{A}_k and $\bar{\mathbf{Y}}_k$ are known.

Step 1. Parameters associated to the objects:

The matrix \mathbf{U}_{k+1} is given by one run of the K-means algorithm on the score matrix $\mathbf{Y}_k = \mathbf{X}\mathbf{A}_k$ starting from the object centroid matrix $\bar{\mathbf{Y}}_k$ in the reduced space. The P new clusters are obtained finding the new centroids, i.e., updating the centroid matrix by $\bar{\mathbf{X}}_{k+1} = (\mathbf{U}_{k+1}^T \mathbf{U}_{k+1})^{-1} \mathbf{U}_{k+1}^T \mathbf{X}$ and the object centroid-based matrix by $\mathbf{Z}_{k+1} = \mathbf{U}_k \bar{\mathbf{X}}_k$.

Every cluster should be assigned with at least one object. On the procedure, if any cluster becomes empty, then a selection step is fulfilled: half of the objects on the bigger cluster is assigned into one of the empty clusters, and this process is repeated while there are empty clusters.

Step 2. Parameters associated to the components:

The updated matrices \mathbf{V}_{k+1} and \mathbf{A}_{k+1} are sequentially constructed row-by-row, and in each row, the process is also sequentially performed column-by-column, in a symbiotic relationship with the maximization of the objective function F .

The matrix \mathbf{V} specifies a partition of the original variables into Q disjoint components. For updating \mathbf{V}_k , each original variable will be evaluated in order to find which component leads to a higher value of the objective function F , assuming that all remaining variables are fixed in the components in accordance with \mathbf{V}_k .

Firstly, the first row of \mathbf{V}_k is updated by detecting for which column j , with $j = 1, \dots, Q$, the allocation of its nonzero element yields better results in the sense of the maximization of the objective function. Concretely, for the first row (variable) of \mathbf{V}_{k+1} , the best column (component) among Q is selected by solving Q PCA subproblems associated to the updated matrices $\mathbf{W}_{k+1}^{(q)}$, for $q = 1, 2, \dots, Q$, respectively, assuming the Q possible positions of the nonzero element into the first row of the potential updated matrix \mathbf{V}_{k+1} . In the q -th PCA subproblem, the first principal component is calculated determining the update of the q -th column of \mathbf{A}_{k+1} . At this point, the centroid matrix on the reduced space, $\bar{\mathbf{Y}}_{k+1}$, and the objective function value, F_{k+1} , can be computed by $\bar{\mathbf{Y}}_{k+1} = \bar{\mathbf{X}}_{k+1} \mathbf{A}_{k+1}$ and $F_{k+1} = \text{tr}((\mathbf{U}_{k+1} \bar{\mathbf{Y}}_{k+1})^T \mathbf{U}_{k+1} \bar{\mathbf{Y}}_{k+1})$. This process is done repeatedly to select the best component to allocate the first row (variable) in \mathbf{V}_{k+1} , which will coincide with the component that yields the highest value of F_{k+1} .

The same process is now repeated for the remaining rows of \mathbf{V}_k , and therefore, \mathbf{V}_{k+1} is updated row-by-row. Hence, for each original variable there are solved Q assignment subproblems. In each subproblem, a subspace of variables is considered and the best direction (eigenvector) with maximum variability explained is obtained performing a PCA step. Each variable will be included into a component associated to the subproblem that maximizes the objective function.

Since there are J original variables, i.e., J rows on \mathbf{V}_k , then there are $J \times Q$ subproblems to be solved in order to obtain \mathbf{V}_{k+1} and \mathbf{A}_{k+1} . At the end of the Step 2, the best assignment will maximize the objective function, and consequently, the between cluster deviance given by $F_{k+1} / \|\mathbf{Y}_{k+1}\|^2$, where $\mathbf{Y}_{k+1} = \mathbf{X} \mathbf{A}_{k+1}$.

Stopping Criterion. Evaluate solutions:

If the difference between F_k and F_{k+1} is smaller than a specified tolerance, then the algorithm stops and returns the current iterates. Otherwise, repeat the iteration, setting $k := k + 1$.

At the end of the algorithm, say, for instance, at the k^* -th iteration, besides returning the allocation matrices \mathbf{U}_{k^*} , for the objects, and \mathbf{V}_{k^*} , for the variables, the component loading matrix \mathbf{A}_{k^*} is also returned, which is a columnwise orthonormal matrix whose elements are the loadings of the CDPCA components. Moreover, the CDPCA component

score matrix \mathbf{Y}_{k^*} is obtained, as well as the object centroid matrix in the reduced space $\bar{\mathbf{Y}}_{k^*}$. These matrices can be used to obtain an approximation of the CDPCA model by

$$\mathbf{U}_{k^*} \bar{\mathbf{Y}}_{k^*} \mathbf{A}_{k^*}^T,$$

providing a partition of the objects along a set of clusters and the variables along a set of disjoint components.

It is worth mentioning that, unlike in the PCA technique, the ALS algorithm can not establish the CDPCA components decreasingly sorted by their explained variability. In order to be consistent with the classical form of representation of the components, at the end of the algorithm the columns of the matrices associated to the CDPCA components, namely, \mathbf{V}_{k^*} , \mathbf{A}_{k^*} , \mathbf{Y}_{k^*} , and $\bar{\mathbf{Y}}_{k^*}$, will be rearranged. Since the changes are performed in all of these matrices, the above CDPCA model is trivially satisfied with the rearranged matrices.

3.3 Synthetic Data

In the following we illustrate an execution of the ALS algorithm described above using the synthetic data. The data matrix is formed by $I = 15$ objects and $J = 3$ variables. In order to evaluate the performance of the algorithm we will also analyse the ability of the algorithm for detecting the $P = 3$ clusters of objects and $Q = 2$ subsets of variables known in the synthetic data.

Considering the synthetic data, we set $k = 0$, specify the convergence tolerance as $\varepsilon = 10^{-5}$, and initialize the parameters of the CDPCA model.

Initialization:

In the Step 1, we get

$$\mathbf{U}_0 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{bmatrix}, \bar{\mathbf{X}}_0 = \begin{bmatrix} 0.690 & -0.416 & -1.022 \\ 0.673 & 0.145 & 0.440 \\ -0.779 & 0.154 & 0.332 \end{bmatrix}, \mathbf{Z}_0 = \begin{bmatrix} 0.690 & -0.416 & -1.022 \\ 0.673 & 0.145 & 0.440 \\ -0.779 & 0.154 & 0.332 \\ 0.690 & -0.416 & -1.022 \\ 0.690 & -0.416 & -1.022 \\ 0.673 & 0.145 & 0.440 \\ 0.690 & -0.416 & -1.022 \\ 0.673 & 0.145 & 0.440 \\ -0.779 & 0.154 & 0.332 \\ -0.779 & 0.154 & 0.332 \\ -0.779 & 0.154 & 0.332 \\ -0.779 & 0.154 & 0.332 \\ 0.673 & 0.145 & 0.440 \\ -0.779 & 0.154 & 0.332 \\ -0.779 & 0.154 & 0.332 \end{bmatrix}$$

In the Step 2, it begins with

$$\mathbf{V}_0 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 0 \end{bmatrix}.$$

Next, we determine \mathbf{A}_0 . Fixing the first column of \mathbf{V}_0 ($q = 1$), the unit normed eigenvector $v_0^{(1)}$ associated to the largest eigenvalue of the correlation matrix of the submatrix $\mathbf{W}_0^{(1)}$ is selected and introduced in the nonzero entries of the first column of \mathbf{A}_0 . A similar procedure is performed for the remaining columns of \mathbf{V}_0 . Thus, for $q = 1$, $K^{(1)} = 2$ and we get the 15×2 matrix

$$\mathbf{W}_0^{(1)} = \begin{bmatrix} 0.690 & -1.022 \\ 0.673 & 0.440 \\ -0.779 & 0.332 \\ 0.690 & -1.022 \\ 0.690 & -1.022 \\ 0.673 & 0.440 \\ 0.690 & -1.022 \\ 0.673 & 0.440 \\ -0.779 & 0.332 \\ -0.779 & 0.332 \\ -0.779 & 0.332 \\ -0.779 & 0.332 \\ 0.673 & 0.440 \\ -0.779 & 0.332 \\ -0.779 & 0.332 \end{bmatrix}.$$

The unit normed eigenvector associated to the largest eigenvalue of the 2×2 matrix $(\mathbf{W}_0^{(1)})^T \mathbf{W}_0^{(1)}$ is given by $v_0^{(1)} = \begin{bmatrix} -0.809 \\ 0.587 \end{bmatrix}$. Hence, for the nonzero elements on the first column of \mathbf{A}_0 , which correspond to the nonzero entries on the first column of \mathbf{V}_0 , we shall introduce $v_0^{(1)}$. Similarly, considering now the second column of \mathbf{V}_0 , we have $q = 2$,

$K^{(2)} = 1$ and $(\mathbf{W}_0^{(2)})^T \mathbf{W}_0^{(2)}$ is a 1×1 matrix. Thus, we get

$$\mathbf{W}_0^{(2)} = \begin{bmatrix} -0.416 \\ 0.145 \\ 0.154 \\ -0.416 \\ -0.416 \\ 0.145 \\ -0.416 \\ 0.145 \\ 0.154 \\ 0.154 \\ 0.154 \\ 0.154 \\ 0.145 \\ 0.154 \\ 0.154 \end{bmatrix} \quad \text{and} \quad v_0^{(2)} = [1],$$

and the nonzero element on the second column of \mathbf{A}_0 will be 1. Therefore, the CDPCA component loading matrix is given by

$$\mathbf{A}_0 = \begin{bmatrix} -0.809 & 0 \\ 0 & 1 \\ 0.587 & 0 \end{bmatrix}.$$

At this point, the objects of the data matrix \mathbf{X} can be assigned in the reduced space of the CDPCA components by the object centroid matrix in the reduced space, $\bar{\mathbf{Y}}_0$, and the objective function F should be evaluated for the current matrices \mathbf{U}_0 and $\bar{\mathbf{Y}}_0$. Regarding

our example, \mathbf{Y}_0 is a 15×2 matrix and $\bar{\mathbf{Y}}_0$ is a 3×2 matrix given as follows.

$$\mathbf{Y}_0 = \begin{bmatrix} -1.621 & -0.981 \\ -1.046 & -1.533 \\ -0.213 & -1.322 \\ -1.316 & -0.425 \\ -1.529 & -1.570 \\ 0.365 & -0.225 \\ -0.172 & 1.310 \\ -0.598 & 0.987 \\ 0.512 & 1.154 \\ 1.561 & 0.036 \\ 0.716 & 1.124 \\ 0.206 & -0.123 \\ 0.132 & 1.353 \\ 1.302 & -0.085 \\ 1.700 & 0.298 \end{bmatrix}, \quad \bar{\mathbf{Y}}_0 = \begin{bmatrix} -1.159 & -0.416 \\ -0.286 & 0.145 \\ 0.826 & 0.154 \end{bmatrix}.$$

initial approximation of \mathbf{Y}_0 provides a partition of objects along a set of three clusters (identified by different colours in the matrix \mathbf{Y}_0 , i.e, objects 1, 4, 5 and 7 are currently assigned into one cluster, objects 2, 6, 8 and 13 are assigned into another cluster, and the remaining objects are currently belonging to a third cluster) and also a partition of variables along a set of disjoint components ($PC_1 = -0.809X_1 + 0.587X_3$ and $PC_2 = X_2$). Notice that the current partition does not correspond to the final solution, nor to the real partition; this is the result after computing the initial step of the CDPCA procedure. Additionally, the objective function value for the current iterates is $F_0 = 11.438$ and the corresponding between cluster deviance is $F_0/\|\mathbf{Y}_0\|_2^2 = 36.63\%$.

First iteration:

Set $k = 1$. In Step 1, the matrix of the allocation of objects into P clusters and the

object centroid matrix are updated yielding the following matrices:

$$\mathbf{U}_1 = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{bmatrix}, \bar{\mathbf{X}}_1 = \begin{bmatrix} 0.735 & -1.166 & -0.936 \\ 0.174 & 0.882 & 0.056 \\ -0.728 & 0.384 & 0.743 \end{bmatrix}, \mathbf{Z}_1 = \begin{bmatrix} 0.735 & -1.166 & -0.936 \\ 0.735 & -1.166 & -0.936 \\ 0.735 & -1.166 & -0.936 \\ 0.735 & -1.166 & -0.936 \\ 0.735 & -1.166 & -0.936 \\ -0.728 & 0.384 & 0.743 \\ 0.174 & 0.882 & 0.056 \\ 0.174 & 0.882 & 0.056 \\ -0.728 & 0.384 & 0.743 \\ -0.728 & 0.384 & 0.743 \\ -0.728 & 0.384 & 0.743 \\ 0.174 & 0.882 & 0.056 \\ 0.174 & 0.882 & 0.056 \\ -0.728 & 0.384 & 0.743 \\ -0.728 & 0.384 & 0.743 \end{bmatrix}$$

Notice that \mathbf{U}_1 specifies a new allocation of the objects.

In Step 2, we get

$$\mathbf{V}_1 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 0 \end{bmatrix}, \mathbf{A}_1 = \begin{bmatrix} -0.660 & 0 \\ 0 & 1 \\ 0.750 & 0 \end{bmatrix},$$

$$\mathbf{Y}_1 = \begin{bmatrix} -1.870 & -0.981 \\ -0.903 & -1.533 \\ -0.186 & -1.322 \\ -1.389 & -0.425 \\ -1.593 & -1.570 \\ 0.682 & -0.225 \\ -0.041 & 1.310 \\ -0.182 & 0.987 \\ 0.799 & 1.154 \\ 1.405 & 0.036 \\ 0.491 & 1.124 \\ -0.011 & -0.123 \\ -0.055 & 1.353 \\ 1.355 & -0.085 \\ 1.502 & 0.298 \end{bmatrix} \text{ and } \bar{\mathbf{Y}}_1 = \begin{bmatrix} -1.188 & -1.166 \\ -0.072 & 0.882 \\ 1.039 & 0.384 \end{bmatrix}.$$

At the end of the first iteration, $F_1 = 24.374$ and the corresponding between cluster deviance is 77.93%. The following step is to check the stopping criterion. Since $|F_1 - F_0| = 12.936 > \varepsilon$, another iteration should be computed.

Further iterations:

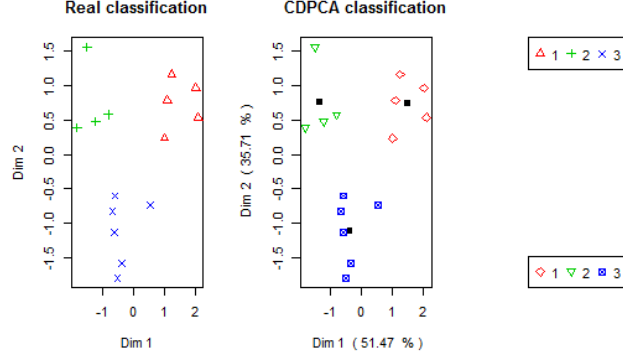


Figure 2: Real and CDPCA classification for the Synthetic Data.

In order to refine the solutions, more iterations of the algorithm are needed. In this example, the best solution was obtained after two iterations and it took only 0.01seconds to exhibit a solution. The obtained results are as follows. The object allocation matrix \mathbf{U} and the variable allocation matrix \mathbf{V} , the component loading matrix \mathbf{A} , the component score matrix \mathbf{Y} and the centroid matrix in the reduced space $\bar{\mathbf{Y}}$ already rearranged by column (in decreasing order of the variability explained by the CDPCA) are given by

$$\mathbf{U} = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{bmatrix}, \quad \mathbf{V} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \\ 1 & 0 \end{bmatrix}, \quad \mathbf{Y} = \begin{bmatrix} 2.092 & 0.535 \\ 1.250 & 1.160 \\ 1.001 & 0.232 \\ 1.101 & 0.782 \\ 2.017 & 0.966 \\ -0.796 & 0.578 \\ -1.211 & 0.478 \\ -1.492 & 1.559 \\ -1.803 & 0.389 \\ -0.352 & -1.581 \\ -0.591 & -1.137 \\ 0.542 & -0.738 \\ -0.584 & -0.603 \\ -0.669 & -0.825 \\ -0.503 & -1.797 \end{bmatrix}.$$

$$\mathbf{A} = \begin{bmatrix} 0 & 1 \\ -0.734 & 0 \\ -0.678 & 0 \end{bmatrix}, \quad \bar{\mathbf{Y}} = \begin{bmatrix} 1.492 & 0.735 \\ -1.325 & 0.751 \\ -0.359 & -1.113 \end{bmatrix},$$

The maximum for the objective function is 31.357 and the corresponding between cluster deviance is 85.63%.

Our R-based implementation of this new methodology provides the graphical display of the CDPCA classification taking the first two CDPCA components, as well as the real classification when it is known. For the synthetic data, the plot is displayed in Figure 2.

Clearly, the CDPCA was able to fulfil the classification and the objects were correctly assigned to the clusters.

Besides that, our R function also returns a pseudo-confusion matrix, here displayed in Table 1. The pseudo-confusion matrix allows one to easily verify how many objects are correctly assigned into clusters, or how many objects are misclassified.

Table 1: Pseudo-confusion matrix for the Synthetic data set.

Real Class	CDPCA Class		
	1	2	3
1	5	0	0
2	0	4	0
3	0	0	6

From Table 1, we can observe that 5 objects are assigned to a cluster, 4 objects are assigned into a second cluster and the remaining 6 objects belong to another cluster. This table confirms the high accuracy classification produced by CDPCA on the Synthetic data.

4 Numerical Experiments

Here, we describe the numerical experiments of the CDPCA applied on a real data set. Our experiments were run on a computer with an Intel Core i5-3317U CPU @ 1.70GHz, with Windows 7 (64 bits) and 6GB RAM, using R version 3.0.0 (2013).

The CDPCA was implemented in R under the function `CDpca` [5]. This function is suitable for data matrices of numeric values.

Since the ALS algorithm can be considered as a heuristic, it is advisable to run the algorithm several times, as it has been suggested in [8], in order to find the global maximum. Therefore, all the presented numerical tests were run 1000 times and the tolerance for convergence purposes was set to 10^{-5} .

Our R implementation of CDPCA starts by standardizing the data. Among other outputs, the `CDpca` function returns the CDPCA component loading matrix, the obtained between cluster deviance, the objects assignment matrix, the variables assignment matrix, a pseudo-confusion matrix when the real classification is known a priori, the variance explained by the CDPCA components and a plot of the data projected into the two dimensional space defined by the first two components is displayed.

4.1 Breast Cancer Data

The Wisconsin Breast Cancer Database [7] contains 683 instances (originally, there were 699 instances; however, 16 of them were excluded since they contain missing values), where each of them is described by 9 attributes with integer values in the range 1 – 10 and a real binary class label, which divides the instances into two classes: benign or malignant.

The list of variables is formed by clump thickness, uniformity of cell size, uniformity of cell shape, marginal adhesion, single epithelial cell size, bare nuclei, bland chromatin, normal nucleoli, and the ninth variable mitoses describes an analysis of mitotic stages. These variables are used in pathology reports for suggesting whether a lump in a breast is benign or malignant.

The CDPCA was applied in this data set, by choosing $P = 2$ clusters of objects and $Q = 2$ subsets of variables and executing our `CDpca` function in R. It took only 6 iterations and 0.19seconds to yield a solution approximation satisfying the convergence tolerance. The results of CDPCA are displayed in Tables 2, 3 and Figure 3.

The Table 2 reports the component loadings for both PCA and CDPCA.

Table 2: Component loadings for PCA and CDPCA on the Breast Cancer Data.

Variables	PCA loadings		CDPCA loadings	
	Component 1	Component 2	Component 1	Component 2
Clump Thickness	-0.296	-0.073	0.350	0
Uniformity of Cell Size	-0.403	0.229	0.429	0
Uniformity of Cell Shape	-0.392	0.164	0.426	0
Marginal Adhesion	-0.331	-0.098	0	-0.710
Single Epithelial Cell Size	-0.249	0.200	0	-0.703
Bare Nuclei	-0.442	-0.780	0.415	0
Bland Chromatin	-0.292	0.008	0.387	0
Normal Nucleoli	-0.354	0.469	0.374	0
Mitoses	-0.124	0.188	0.216	0
Explained Variance(%)	69.05	7.20	51.71	17.74

Comparing the results in Table 2, performing an analysis of data from the obtained results using the PCA technique can be complex. The resulting PCA component loadings lead to components which do not seem interpretable. This is due to all the original variables contribute to both PCA components and, therefore, it is quite difficult to detect a pattern or relation among the variables for each of the two first principal components. With CDPCA the interpretation of the components becomes easier, since each variable contributes to a single component.

The first PCA component explains 69,05% of the total variance and is mainly characterized by Bare Nuclei, Uniformity of Cell Size and Uniformity of Cell Shape, while the second PCA component explains only 7,20% of the total variance and is mainly characterized by Bare Nuclei and Normal Nucleoli. Notice that the variable Bare Nuclei is the most contributing variable for both components.

Considering now the CDPCA technique, it can be observed that the first CDPCA component explains 51,71% of the total variance and is mainly characterized by Uniformity of Cell Size, Uniformity of Cell Shape and Bare Nuclei, while the second CDPCA component is only characterized by the original variables Marginal Adhesion and Single

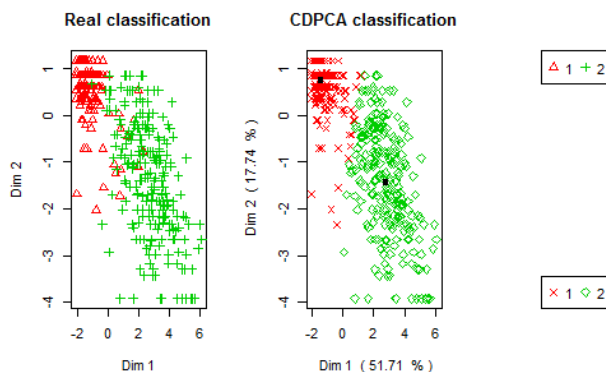


Figure 3: Real and CDPCA classification for the Breast Cancer Data.

Epithelial Cell Size, explaining 17,74% of the total variance.

Table 3: Pseudo-confusion matrix for the Breast Cancer Data.

Real class	Predictive CDPCA class	
	1 (benign)	2 (malignant)
1 (benign)	434	10
2 (malignant)	19	220

Table 3 evaluates the predictive performance of CDPCA as a classification technique on the Breast Cancer data. The real classification for this data set is as follows: 444 objects into the benign class, and 239 into the malignant. Considering the pseudo-confusion matrix obtained with the results on the CDPCA classification, we conclude that 453 objects are assigned to the benign class and 230 are included into the malignant class. This means that there are 29 misclassified objects, leading to a 4% of misclassification. Therefore, the CDPCA classification presents an accuracy of 96% permitting to conclude that our implementation of the CDPCA performed very well in practice.

In Figure 3, a graph representation of the data into the 2-dimensional reduced space defined by the first two CDPCA components is depicted. This graph permits to visualize the data in order to help on the detection of patterns hidden in the data set. In the case of the Breast Cancer data, the graph shows that positive value for the first CDPCA component is tendentially attributed to subjects (objects) with malignant lumps (class 2).

The obtained CDPCA between cluster deviance is 80,20% of the total deviance.

5 Conclusions

Applications of the recently developed methodology CDPCA to data reveal that this method can be successful for classifying the samples and exploring relationship between variables, as well as for visualizing data into a reduced space. This paper is particularly focussed on detailing a two-step-based scheme of the ALS algorithm used to perform CDPCA and on its algebraic features. A toy example is included to illustrate the resulting transformations on the ALS algorithm step by step. A final remark is that the ALS algorithm for CDPCA performed very well and also revealed high accuracy in the clusterings for the presented examples and several other not shown herein.

Acknowledgments.

The authors would like to thank the anonymous referee for all the valuable and constructive comments which have helped to improve this paper. A special thanks to Professor Maurizio Vichi for providing us a Matlab version of the ALS algorithm for performing CDPCA. This work was partially supported by Portuguese funds through the CIDMA - Center for Research and Development in Mathematics and Applications, and the Portuguese Foundation for Science and Technology (FCT – Fundação para a Ciência e a Tecnologia), within project UID/MAT/04106/2013.

References

- [1] d’Aspremont, A., El Ghaoui, L., Jordan, M.I., Lanckriet, G.R.G.: A Direct Formulation for Sparse PCA Using Semidefinite Programming. *SIAM* 49(3), pp. 434–448 (2007)
- [2] Jolliffe, I.T.: *Principal Component Analysis*. Second edition, Springer-Verlag, New York (2002)
- [3] Jolliffe, I.T., Trendafilov, N.T., Uddin, M.: A modified principal component technique based on the lasso. *J. of Computational and Graphical Statistics* 12(3), pp. 531–547 (2003)
- [4] Ma, Z.: Sparse principal component analysis and iterative thresholding. *The Annals of Statistics* 41(2), pp. 772–801 (2013)
- [5] Macedo, E., Freitas, A.: Statistical Methods and Optimization in Data Mining. In: *III International Conference of Optimization and Applications OPTIMA2012*, pp. 164–169 (2012)
- [6] R Development Core Team: *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, <http://www.R-project.org/>

- [7] UCI Repository: Winsconsin Breast Cancer Data Set. [http://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+\(Original\)](http://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+(Original))
- [8] Vichi, M., Saporta, G.: Clustering and Disjoint Principal Component Analysis. *Computational Statistics & Data Analysis* 53, pp. 3194–3208 (2009)
- [9] Vines, S.: Simple principal components. *Applied Statistics* 49, pp. 441–451 (2000)
- [10] Xu, R., Wunsch, D.: Survey of Clustering Algorithms. *IEEE Transactions on Neural Networks* 16, pp. 645–648 (2005)
- [11] Zou, H., Hastie, T., Tibshirani, R.: Sparse principal component analysis. *J. of Computational and Graphical Statistics* 15(2), pp. 262–286 (2006)