Principal Component Analysis with Boundary Constraints

PAOLO GIORDANI
Department of Statistics, Probability and Applied Statistics
Sapienza University of Rome
P.le Aldo Moro, 5, 00185 Rome, Italy
paolo.giordani@uniroma1.it

HENK A.L. KIERS
Heymans Institute (DPMG)
University of Groningen
Grote Kruisstraat 2/1, 9712 TS Groningen, The Netherlands
h.a.l.kiers@rug.nl

Abstract: Observed data often belong to some specific intervals of values (for instance in case of percentages or proportions) or are higher (lower) than pre-specified values (for instance, chemical concentrations are higher than zero). The use of classical Principal Component Analysis (PCA) may lead to extract components such that the reconstructed data take unfeasible values. In order to cope with this problem, a constrained generalization of PCA is proposed. The new technique, called Bounded Principal Component Analysis (B-PCA), detects components such that the reconstructed data are constrained to belong to some pre-specified bounds. This is done by implementing a row-wise Alternating Least Squares algorithm, which exploits the potentialities of the Least Squares with Inequality (LSI) algorithm. The results of a simulation study and two applications to bounded data are discussed for evaluating how the method and the algorithm for solving it work in practice.

Keywords: Alternating Least Squares procedure, Non-Negative Least Squares algorithm, Least Squares with Inequality algorithm, Principal Component Analysis, Bounded data.

1. Introduction

Alternating Least Squares (ALS) procedures have been extensively used in component analysis. ALS procedures solve a least squares problem with respect to a set of parameters. The intuition behind ALS techniques is to split the set of the parameters into several subsets and to determine iteratively every optimal subset of parameters conditionally upon considering the other subsets as fixed. Such an iterative procedure is repeated until the value of the function to be minimized does not vary noticeably across two consecutive iterations. As the loss function is bounded and at every iteration a least squares problem is solved, the loss function value decreases or remains the same. This guarantees the convergence of the procedure.

In the literature, several constrained versions of standard Alternating Least Squares (ALS) algorithms have been introduced (see, for instance, [1-12]).

Most of the previous mentioned works are devoted to suitable constrained generalizations of Principal Component Analysis (PCA). PCA is a well-known tool aiming at detecting the underlying structure of a data set stored in matrix **X**. Specifically, PCA consists of synthesizing the (n×m)-matrix **X** as

$$\mathbf{X} \cong \mathbf{A}\mathbf{B}'. \qquad (1)$$

In (1), **X** is summarized by matrix **AB′** having rank p ($<m$), where **A** (n×p) and **B** (m×p) are, respectively, the component score and component loading matrices and p is the number of extracted components. Usually, n and m denote the numbers of observation units and variables, respectively. The optimal component matrices are obtained by minimizing

$$||\mathbf{X}\text{-}\mathbf{A}\mathbf{B}'||^2. \qquad (2)$$

As is known, the *unconstrained* minimization of (2) can be attained by means of the Singular Value Decomposition of **X**. Furthermore, the loss function in (2) can be minimized according to various constraints concerning the component matrices. Generally speaking, for instance, one can set these matrices to be equal to pre-specified ones or linearly constrain them by means of external information. A different way to act consists of requiring that certain elements are equal to zero, to a non-zero pre-specified value, to each other or are non-negative. The latter case is useful when negative component scores make no sense from an empirical point of view.

In several real life applications, the data set at hand may refer to phenomena described by variables or attributes the values of which belong to a given domain of scores. For instance, in spectroscopic data, the scores are non-negative and, at the same time, it can be reasonable to assume that these are lower than a maximum value. The same comment holds for concentrations of chemical substances. Negative values are unfeasible as well as extremely high concentrations. Moreover, one may think about sensory data which contains the ratings of a set of judges on a collection of goods. The ratings may range from a lower bound (not necessarily equal to zero) to an upper bound. Once again, in case of proportions or percentages, the data must belong to the interval [0,1] or [0,100], respectively. In all of these situations, it would be desirable that the reconstructed data fulfil the requirement that they are higher than the lower bounds of the intervals to which they belong and lower than the upper bounds. In other words, it can be useful to get feasible reconstructed data without imposing any constraint on the component matrices.

To corroborate our claim, let us consider the Household data set [13], which refers to the percentage of households in each of 16 European countries who consume a specific food items among a set of 20 food items. The aim of the study is to investigate the existence of possible (dis)similarities among European countries in the food consumption. Unfortunately, three data elements are missing. We thus decided to ignore the associated countries. We have then examined the information concerning n = 13 countries and m = 20 food items. Following [13], we have performed (unconstrained) PCA extracting p = 2 components and prior to fitting the model, we have preprocessed the data by centering and scaling (see also Section 3.3, in the sequel). The fit percentage using p = 2 components was equal to 50.22%. By inspecting the reconstructed data (applying the inverse preprocessing procedure), we saw that seven figures took unfeasible values, i.e. outside the interval [0, 100]. In particular, four times we got reconstructed percentages lower than 0%. Three times out of four, this occurred for percentages concerning Tinned Soup with respect to Italy (observed datum = 3% and reconstructed datum = -1.8%), Austria (1% and -0.4%, respectively) and, above all, Portugal (1% and -11.0%, respectively). Finally, the observed percentage concerning the consumption of frozen fish in Ireland is equal to 5%, but the reconstructed score was -2.1%. Regarding unfeasible scores higher than 100%, we can mention the consumption of Ground Coffee (observed 96% and reconstructed 108.3%) and Margarine (observed 91% and reconstructed 110.4%) by Danish households. Finally, as one would expect, the observed consumption of tea in England is very high (99%). However, this does not admit a reconstructed consumption equal to 100.5%. Note that, if we chose p = 3 components (the fit percentage was 65.19%), we got seven unfeasible values. Three of them were already found using p = 2 components. Going into detail, these are the percentages of Tinned Soup in Italy (-3.4%) and Portugal (-9.5%) and Tea in England (106.4%). Moreover, we found reconstructed percentages of Tinned Soup in Luxembourg equal to 105.2% (observed 97%) and of Jam and Garlic in England equal to 100.1% and –9.9%, respectively (observed 91% and 11% respectively). These results seem to show that the increase of the number of extracted components does not prevent the tendency of PCA to obtain unfeasible values.

Summing up, we think that the components to be extracted should be found in such a way that the reconstructed data take feasible values, otherwise, if at least one reconstructed score is unfeasible, the extracted components are somehow meaningless. It is thus convenient to constrain the model in order to ensure that the obtained components determine *feasible* reconstructed data.

The paper is organized as follows. In the next section, three linear least squares problems with linear inequality constraints (Non-Negative Least Squares (NNLS), Least Distance Programming (LDP), and Least Squares with Inequality (LSI) problems) are briefly recalled [14]. In section 3, the constrained PCA problem is formalized and the iterative solution based on the LSI algorithm is described. Section 4 is devoted to a simulation experiment in order to compare the abilities in recovering the component structure underlying the data of the proposed constrained PCA and ordinary PCA and to evaluate the performance of the proposed algorithm in terms of computation time and risk of hitting local optima. Finally, extensive applications of the constrained PCA method to the Household data set and to a chemical data set are given in Section 5.

2. Linear least squares with linear inequality constraints

2.1. The Non-Negative Least Squares (NNLS) problem

Let **U** be an $(n \times m)$-matrix, **v** an n-vector and **w** an m-vector. The standard NNLS problem consists of

$$\begin{aligned} &\min_{\mathbf{w}} \|\mathbf{Uw} - \mathbf{v}\|^2, \\ &\text{s.t. } \mathbf{w} \geq \mathbf{0}_m, \end{aligned} \tag{3}$$

where $\mathbf{0}_m$ is an m-vector with zero elements. A detailed description of the NNLS algorithm along with the proofs of the optimality of the obtained solution and of the convergence of the algorithm in a finite number of iterations can be found in [14]. Also note that modifications of the NNLS algorithm for reducing the computation time have been proposed in [4, 12].

2.2. The Least Distance Programming (LDP) problem

A different problem, usually referred to as Least Distance Programming (LDP) can be written as follows:

$$\begin{aligned}&\min_{\mathbf{w}} \|\mathbf{w}\|^2,\\ &\text{s.t. } \mathbf{Gw} \geq \mathbf{h},\end{aligned} \tag{4}$$

where **G** is a $(l \times m)$-matrix and **h** an *m*-vector. The LDP problem in (4) consists of first assessing whether $\mathbf{Gw} \geq \mathbf{h}$ is consistent and, if so, of finding the best value of **w** in the sense that the loss function in (4) is minimized. The optimal solution of (4) can be found by transforming the minimization problem into a particular NNLS problem. See, for more details, [14].

2.3. The Least Squares with Inequality (LSI) problem

The NNLS problem given in (3) can be also generalized by means of the constraints of the LDP problem in (4). Then, the problem to be solved can be expressed as

$$\begin{aligned}&\min_{\mathbf{w}} \|\mathbf{Uw} - \mathbf{v}\|^2,\\ &\text{s.t. } \mathbf{Gw} \geq \mathbf{h}.\end{aligned} \tag{5}$$

In the literature, this problem is usually christened as Least Squares with Inequality (LSI). The optimal solution of (5) can be attained by observing that the LSI problem can be transformed into a particular LDP problem. To this purpose, we can decompose **U** (having rank *m*) by means of the SVD. Let $\mathbf{U} = \mathbf{P}\begin{bmatrix}\mathbf{Q}\\ \mathbf{0}_{(n-m)\times m}\end{bmatrix}\mathbf{R}'$, where **P** (*n*×*n*) and **R** (*m*×*m*) contain the orthonormal left and right singular vectors of **U**, **Q** (*m*×*m*) is a diagonal matrix containing the singular values (of **U**) and $\mathbf{0}_{(n-m)\times m}$ is a $((n-m)\times m)$ matrix with zero elements. If we define **y** = **R′w**, hence **w** = **Ry**, then (5) can be rewritten as

$$\begin{aligned}&\min_{\mathbf{y}} \left\| \begin{bmatrix}\mathbf{P}_1'\\ \mathbf{P}_2'\end{bmatrix}\mathbf{v} - \begin{bmatrix}\mathbf{Qy}\\ \mathbf{0}_{(n-m)\times m}\end{bmatrix} \right\|^2,\\ &\text{s.t. } \mathbf{GRy} \geq \mathbf{h},\end{aligned} \tag{6}$$

where $\mathbf{P}_1$ (*n*×*m*) and $\mathbf{P}_2$ (*n*×*n*-*m*) contain the first *n* and last *n*-*m* columns of **P**. By setting $\mathbf{s} = \mathbf{Qy} - \mathbf{P}_1'\mathbf{v}$, (6) can be expressed as

$$\begin{aligned}&\min_{\mathbf{s}} \|\mathbf{s}\|^2 + \|\mathbf{P}_2'\mathbf{v}\|^2,\\ &\text{s.t. } \mathbf{GRQ}^{-1}\mathbf{s} \geq \mathbf{h} - \mathbf{GRQ}^{-1}\mathbf{P}_1'\mathbf{v}.\end{aligned} \tag{7}$$

The problem in (7) coincides with that in (4) after observing that $\|\mathbf{P}_2'\mathbf{v}\|^2$ can be considered as a constant with respect to **s**. Of course, it follows that the LDP algorithm can be adopted for solving (7).

3. Bounded Principal Component Analysis (B-PCA).

3.1. Problem

Let **X** (n×m) be the observed data matrix such that $\underline{\mathbf{X}} \leq \mathbf{X} \leq \overline{\mathbf{X}}$, where $\underline{\mathbf{X}}$ and $\overline{\mathbf{X}}$ are the (n×m) matrices of the lower and upper bounds, respectively. We now aim at summarizing matrix **X** under the constraints that the reconstructed data **AB**' must belong to the interval $\left[\underline{\mathbf{X}}, \overline{\mathbf{X}}\right]$. Therefore, the problem can be formalized as

$$\begin{aligned}&\min_{\mathbf{A},\mathbf{B}} \|\mathbf{X} - \mathbf{AB}'\|^2,\\ &\text{s.t. } \underline{\mathbf{X}} \leq \mathbf{AB}' \leq \overline{\mathbf{X}}.\end{aligned} \tag{8}$$

We refer to the constrained PCA problem in (8) as Bounded Principal Component Analysis (B-PCA).

3.2. Solution

The optimal solution of the minimization problem in (8) can be obtained by performing an Alternating Least Squares (ALS) algorithm involving the solutions of specific LSI problems. In fact, as we shall see, during each iteration we update the rows of **A** and **B** by solving ad-hoc LSI problems. When we update with respect to a row, we solve the minimization problem in (8) keeping fixed the other rows of the component matrices. Whenever a row is updated, the loss function to be minimized decreases. After updating all the rows, if the value of the loss function decreases less than a specified value, say ε, (or than a specified percentage from the previous function value), we consider the algorithm converged, otherwise we repeat the updating of all the rows. As the expression in (8) has a lower bound, the function value converges to a stable value. This guarantees that the algorithm converges to, at least, a local optimum. To limit the risk of hitting local optima, more than one random start is recommended.

Let us now describe in detail how to update the rows of **A** and **B**. Let us consider the update of the generic i-th row of **A**, say $\mathbf{a}_i'$, i = 1,…,n. The loss function in (8) can be rewritten as

$$\min_{\mathbf{a}_i} \|\mathbf{x}_i' - \mathbf{a}_i'\mathbf{B}'\|^2 + \|\mathbf{X}_{(i)} - \mathbf{A}_{(i)}\mathbf{B}'\|^2, \tag{9}$$

where $\mathbf{x}_i'$, i = 1,…,n, denotes the i-th row of **X**, and $\mathbf{X}_{(i)}$ and $\mathbf{A}_{(i)}$ denote the matrices **X** and **A** with the i-th row deleted, respectively. The second term of (9) can be considered as a constant with respect to $\mathbf{a}_i'$ and will be thus be ignored in the sequel. Therefore, it follows that (9) can be replaced by

$$\min_{\mathbf{a}_i} \left\| \mathbf{B}\mathbf{a}_i - \mathbf{x}_i \right\|^2 . \tag{10}$$

Furthermore, from (8), the update of $\mathbf{a}_i'$ must fulfil the requirement that

$$\underline{\mathbf{x}}_i' \leq \mathbf{a}_i'\mathbf{B}' \leq \bar{\mathbf{x}}_i', \tag{11}$$

where $\underline{\mathbf{x}}_i'$ and $\bar{\mathbf{x}}_i'$ denote the *i*-row of $\underline{\mathbf{X}}$ and $\bar{\mathbf{X}}$, respectively. After some manipulations, it is easy to see that (11) can be also expressed as

$$\begin{matrix} \mathbf{B}\mathbf{a}_i \geq \underline{\mathbf{x}}_i \\ -\mathbf{B}\mathbf{a}_i \geq -\bar{\mathbf{x}}_i \end{matrix} \Leftrightarrow \begin{bmatrix} \mathbf{B} \\ -\mathbf{B} \end{bmatrix} \mathbf{a}_i \geq \begin{bmatrix} \underline{\mathbf{x}}_i \\ -\bar{\mathbf{x}}_i \end{bmatrix}. \tag{12}$$

By combining (10) and (12), we get that the optimal row $\mathbf{a}_i'$ can be found by solving the following problem:

$$\begin{aligned} &\min_{\mathbf{a}_i} \left\| \mathbf{B}\mathbf{a}_i - \mathbf{x}_i \right\|^2 , \\ &\text{s.t.} \begin{bmatrix} \mathbf{B} \\ -\mathbf{B} \end{bmatrix} \mathbf{a}_i \geq \begin{bmatrix} \underline{\mathbf{x}}_i \\ -\bar{\mathbf{x}}_i \end{bmatrix}. \end{aligned} \tag{13}$$

The solution of (13) can be obtained taking into account that the problem in (13) coincides with the LSI problem in (5) setting $\mathbf{U} = \mathbf{B}$, $\mathbf{w} = \mathbf{a}_i$, $\mathbf{v} = \mathbf{x}_i$, $\mathbf{G} = \begin{bmatrix} \mathbf{B} \\ -\mathbf{B} \end{bmatrix}$ and $\mathbf{h} = \begin{bmatrix} \underline{\mathbf{x}}_i \\ -\bar{\mathbf{x}}_i \end{bmatrix}$.

Analogously, we can determine the minimization problem for updating the *j*-th row of **B**, say $\mathbf{b}_j'$, *j* = 1,…,*m*. In fact, bearing in mind that the other rows of the component matrices can be considered as fixed, the function to be minimized reduces to

$$\min_{\mathbf{b}_j} \left\| \mathbf{A}\mathbf{b}_j - \mathbf{x}^j \right\|^2 , \tag{14}$$

where $\mathbf{x}^j$, *j* = 1,…,*m*, denotes the *j*-th column of **X**. The constraints in (8) regarding $\mathbf{b}_j'$ can be written as

$$\begin{matrix} \mathbf{A}\mathbf{b}_j \geq \underline{\mathbf{x}}^j \\ -\mathbf{A}\mathbf{b}_j \geq -\bar{\mathbf{x}}^j \end{matrix} \Leftrightarrow \begin{bmatrix} \mathbf{A} \\ -\mathbf{A} \end{bmatrix} \mathbf{b}_j \geq \begin{bmatrix} \underline{\mathbf{x}}^j \\ -\bar{\mathbf{x}}^j \end{bmatrix}, \tag{15}$$

where $\underline{\mathbf{x}}^j$ and $\bar{\mathbf{x}}^j$ denote the *j*-column of $\underline{\mathbf{X}}$ and $\bar{\mathbf{X}}$, respectively. Expressions (14) and (15) yield

$$\begin{aligned} &\min_{\mathbf{b}_j} \left\| \mathbf{A}\mathbf{b}_j - \mathbf{x}^j \right\|^2 , \\ &\text{s.t.} \begin{bmatrix} \mathbf{A} \\ -\mathbf{A} \end{bmatrix} \mathbf{b}_j \geq \begin{bmatrix} \underline{\mathbf{x}}^j \\ -\bar{\mathbf{x}}^j \end{bmatrix}, \end{aligned} \tag{16}$$

which coincides with the LSI problem in (5) setting $\mathbf{U} = \mathbf{A}$, $\mathbf{w} = \mathbf{b}_j$, $\mathbf{v} = \mathbf{x}^j$, $\mathbf{G} = \begin{bmatrix} \mathbf{A} \\ -\mathbf{A} \end{bmatrix}$ and $\mathbf{h} = \begin{bmatrix} \underline{\mathbf{x}}^j \\ -\overline{\mathbf{x}}^j \end{bmatrix}$.

Summing up, the following row-wise constrained ALS algorithm can be implemented for solving the B-PCA problem introduced in (5).

```
Step 0   Fix $\varepsilon$ and $p$ (number of extracted components).
         Construct, for instance randomly, component matrices $\mathbf{A}^{[0]}$ and $\mathbf{B}^{[0]}$.
         Compute $f^{[0]} = \left\| \mathbf{X} - \mathbf{A}^{[0]}\mathbf{B}^{[0]\prime} \right\|^2$.
         Set $k = 1$.
Step 1   Update $\mathbf{a}_i'^{[k]}$, $i = 1,\ldots,n$, by solving (13).
Step 2   Update $\mathbf{b}_j'^{[k]}$, $j = 1,\ldots,m$, by solving (16).
Step 3   Compute $f^{[k]} = \left\| \mathbf{X} - \mathbf{A}^{[k]}\mathbf{B}^{[k]\prime} \right\|^2$.
Step 4   If $\left| f^{[k-1]} - f^{[k]} \right| < \varepsilon$, then the algorithm has converged; otherwise set $k = k + 1$ and go to
         Step 1.
```

3.3. Preprocessing

In several occasions, the data at hand need to be preprocessed. This can be done both by centering in order to eliminate the so-called offset terms and by scaling in order to eliminate artificial differences among the variables. The centering procedure can be performed by subtracting the mean value of every variable from the observed scores. We then get the centered data **QX** where $\mathbf{Q} = \mathbf{I}_n - \frac{\mathbf{1}_{n \times n}}{n}$ is the centering operator, in which $\mathbf{I}_n$ is the identity matrix of order *n* and $\mathbf{1}_{n \times n}$ is an (*n*×*n*) matrix with unit elements. Using a different notation, the centered data can be also written as $\mathbf{X} - \mathbf{1}_n \mathbf{m}'$, where **m** is the column vector of order *m* holding the mean of the *j*-th variable in the *j*-th element and $\mathbf{1}_n$ is a row vector of order *n* with unit elements. The scaling procedure consists of multiplying the scores of every variable by a scalar, which is usually the inverse of the standard deviation. We thus obtain the scaled data **XD**, where **D** is the diagonal matrix of order *m* with the inverses of the standard deviations as non-zero elements. In case of both centering and scaling, PCA is performed on matrix **QXD**. In order to guarantee that the constraints in (8) are still satisfied after preprocessing (applying the inverse preprocessing procedure), it is necessary to preprocess the lower and upper bounds using the same transformations adopted for the data matrix **X**. Therefore, the B-PCA problem in (8) is replaced by

$$\begin{aligned}&\min_{\mathbf{A},\mathbf{B}} \left\|\mathbf{QXD}-\mathbf{AB}'\right\|^2\\ &\text{s.t. } \left(\underline{\mathbf{X}}-\mathbf{1}_n\mathbf{m}'\right)\mathbf{D} \le \mathbf{AB}' \le \left(\overline{\mathbf{X}}-\mathbf{1}_n\mathbf{m}'\right)\mathbf{D}.\end{aligned} \tag{17}$$

It should be clear that the problem in (17) can be minimized by iteratively solving the LSI problems in (13) and (16) replacing the rows and the columns of **X**, $\underline{\mathbf{X}}$ and $\overline{\mathbf{X}}$ with the preprocessed ones implicitly defined in (17).

Remark 1: Rotational freedom

As for classical unconstrained PCA, the B-PCA method allows rotations of the obtained components. In fact, if the component matrices **A** and **B** satisfy the membership of the reconstructed data in the interval $\left[\underline{\mathbf{X}},\overline{\mathbf{X}}\right]$ and **T** is a (*p*×*p*) rotation matrix, then $\tilde{\mathbf{A}}=\mathbf{AT}$ and $\tilde{\mathbf{B}}=\mathbf{BT}'^{-1}$ are feasible component matrices because $\tilde{\mathbf{A}}\tilde{\mathbf{B}}'=\mathbf{AT}\left(\mathbf{BT}'^{-1}\right)'=\mathbf{ATT}^{-1}\mathbf{B}'=\mathbf{AB}'\in\left[\underline{\mathbf{X}},\overline{\mathbf{X}}\right]$. Obviously, this can be useful for simple structure rotations in order to facilitate the interpretation of the extracted components.

Remark 2: Non-Negative Principal Component Analysis

Suppose that the data are bounded to be non-negative ($\underline{\mathbf{X}}=\mathbf{0}_{n\times m}$ where $\mathbf{0}_{n\times m}$ is a $(n\times m)$ matrix with zero elements and $\overline{\mathbf{X}}$ is ignored) and the centering step is not required. Without loss of generality, let us assume that the data matrix **X** holds scaled scores. The B-PCA problem reduces to

$$\begin{aligned}&\min_{\mathbf{A},\mathbf{B}} \left\|\mathbf{X}-\mathbf{AB}'\right\|^2,\\ &\text{s.t. } \mathbf{AB}' \ge \mathbf{0}_{n\times m},\end{aligned} \tag{18}$$

which strongly resembles the so-called Non-Negative Principal Component Analyses (NN-PCA) (see, e.g., [15]). NN-PCA, which can be seen as a particular case of the so-called archetypal analysis proposed in [16], is formalized as

$$\begin{aligned}&\min_{\mathbf{A},\mathbf{B}} \left\|\mathbf{X}-\mathbf{AB}'\right\|^2,\\ &\text{s.t. } \mathbf{A} \ge \mathbf{0}_{n\times p}, \mathbf{B} \ge \mathbf{0}_{m\times p},\end{aligned} \tag{19}$$

where $\mathbf{0}_{n\times p}$ and $\mathbf{0}_{m\times p}$ are matrices with zero elements of order $(n\times p)$ and $(m\times p)$, respectively. Even if, from a mathematical point of view, problems (18) and (19) differ and have different solutions, from a practical point of view, the differences seem to be negligible.

4. Results on simulated data

In order to evaluate the performance of the B-PCA method, a simulation experiment was carried out. The research questions to be answered by means of the simulation study are:

- What is the recovering performance of B-PCA compared to that of ordinary PCA?
- What is the tendency of the B-PCA algorithm to hit local optima?
- Is the B-PCA algorithm efficient?
- Is it preferable to consider the PCA solution as a rational starting point?

To answer the above questions, data sets with a known underlying structure were randomly generated and noise was added. In particular, the generated data had six different data sizes with the numbers of observation units (n) and variables (m) ranging from 10 to 50 (10×10, 30×10, 30×30, 50×10, 50×30 and 50×50). The data stored in the (n×m)-matrix **X** were constructed according to

$$\mathbf{X} = \mathbf{AB}' + \eta\mathbf{N}, \tag{20}$$

where **A** and **B** are the known component matrices and **N** is the noise matrix; to quantify exactly the relative amount of noise, the parameter η, which takes values 0.0, 0.1, 0.5 and 1.0 was introduced and the noise matrices were scaled in such a way that $\|\mathbf{AB}'\|^2=\|\mathbf{N}\|^2$. All these three matrices were generated by two different approaches. Specifically, in the *uniform* case, their elements were randomly generated from the uniform distribution in [-1,1], whereas, in the *normal* case, from the standard normal distribution. The numbers of columns of **A** and **B**, e.g. the numbers of components, were chosen equal to p=2 or 3. For every level of every design variable, five data sets were randomly generated. Therefore, 6 (data sizes) × 2 (*uniform* or *normal* cases) × 2 (numbers of components) × 4 (levels of noise) × 5 (replications) = 480 data sets were randomly generated during the simulation experiment.

We assumed that the lower and upper bounds of every randomly generated data matrix were, for every column, the minimal and maximal observed values, respectively. Thus, we get $\underline{\mathbf{X}} = \mathbf{1}_n \underline{\mathbf{m}}$ and $\bar{\mathbf{X}} = \mathbf{1}_n \bar{\mathbf{m}}$, where $\underline{\mathbf{m}}$ and $\bar{\mathbf{m}}$ are vectors of size m, in which the generic j-th elements correspond to the minimum and maximum of the j-th column of **X**, respectively. Therefore, we have randomly generated data matrices in a somewhat special way, because they contains (at least) 2m elements which coincide with the extreme values.

B-PCA and ordinary PCA were then performed on the randomly generated data sets. We decided to extract two or three components according to the level of the corresponding design variable p for the data set at hand. For each case, eleven different starts of the B-PCA algorithm have been considered. Specifically, the algorithm was run using one rational start based on the PCA solution of **X**, and ten random starts. For the random starts, the component matrices have been randomly generated from the same distribution as the matrices used in generating the data, i.e, in the *uniform* case, from U[-1,1] and, in the *normal* case, from N(0,1).

In order to compare the performances of B-PCA and ordinary PCA, we have checked whether B-PCA is better in recovering **AB'** than is PCA. We have chose the Proportion of Agreement (PA) index (see, e.g, [17]) for evaluating how B-PCA and PCA worked. With respect to B-PCA, we then get:

$$PA = \frac{\left\|\mathbf{A}_{B-PCA}\mathbf{B}'_{B-PCA} - \mathbf{AB}'\right\|}{\left\|\mathbf{AB}'\right\|} \times 100, \tag{21}$$

where $\mathbf{A}_{B\text{-}PCA}$ and $\mathbf{B}_{B\text{-}PCA}$ are the optimal component matrices resulting from B-PCA (considering, for each case, the values pertaining to the run of the algorithm that led to the smallest function value). The PA index for PCA can be obtained analogously for the optimal component matrices from ordinary PCA. The PA index takes values from 0 to 100. When the score of the index is 100, the method at hand perfectly recovers the component structure. Table 1 contains the average PA values distinguishing among the levels of the design variables.

Table 1. *Recovering performances of B-PCA and PCA: average PA values, mean PA differences ($PA_{B\text{-}PCA}$- PA_{PCA}) and 95% confidence intervals in brackets. The values are distinguished with respect to the data size, the generating distribution, the level of noise and the number of components.*

Design Variable	PCA	B-PCA	mean PA differences ($PA_{B\text{-}PCA}$- PA_{PCA}) and [95% Confidence Intervals]
10×10	82.47	82.94	0.47 [0.07, 0.86]
30×10	87.98	88.18	0.20 [0.08, 0.31]
30×30	94.22	94.23	0.01 [-0.04, 0.05]
50×10	89.92	89.99	0.07 [0.02, 0.13]
50×30	95.50	95.51	0.01 [-0.02, 0.04]
50×50	96.72	96.71	-0.01 [-0.04, 0.03]
N(0,1)	91.20	91.34	0.14 [0.04, 0.24]
U[-1,1]	91.07	91.18	0.11 [0.01, 0.21]
η=0.0	100.00	100.00	0.00 [0.00, 0.00]
η=0.1	99.76	99.71	-0.05 [-0.06, -0.04]
η=0.5	93.71	93.67	-0.04 [-0.11, 0.03]
η=1.0	71.06	71.66	0.60 [0.34, 0.85]
p=2	92.88	92.98	0.10 [-0.01, 0.21]
p=3	89.39	89.54	0.15 [0.06, 0.24]
Overall	91.13	91.26	0.13 [0.05, 0.20]

From Table 1, we can observe that B-PCA worked at least equally well or better than ordinary PCA with respect to the recovery of the component parameters. In fact, the average PA value concerning B-PCA is slightly higher than that concerning PCA (91.26% for B-PCA and 91.13% for PCA) and the associated 95% confidence interval for the mean PA differences does not contain 0 and has positive bounds. It can be seen that this also holds for many levels of the design variables (notice that the best B-PCA performance is registered for case 'η=1.0'). In some cases, confidence intervals around mean PA differences contain 0, and hence apparently the mean difference is not clearly and reliably systematically positive in favour of either method. Among all the levels of all the design variables, just once, when η=0.1, PCA seemed to work better than B-PCA. All in all, we can therefore conclude that the use of B-PCA does not only ensure that we get estimates within the given bounds, but also gives solutions that are typically better and never clearly worse than PCA in recovering the underlying structure.

These results showed the usefulness of B-PCA, as compared to PCA and encouraged to give a better insight into its potentialities. Among them, the tendency of the B-PCA algorithm to hit local optima has been studied by checking for each case, the percentage of times in which the function value was more than 0.1% bigger than that corresponding to the optimal solution, i.e the lowest function value obtained in the eleven runs. This was done for η≥0.1. On the contrary, when η=0.0, since the optimal function value is equal to 0 (i.e., the model perfectly fits the data), we assumed that the global optimum has been attained if the function value was lower than 10^{-6}. The results are displayed in Table 2.

Table 2. *Percentages of times in which the global optimum was attained. The values are distinguished with respect to the data size, the level of noise, the number of components, the generating distribution and the kind of start.*

Design Variable	B-PCA		
	Rational Start	Random Start	Rational+Random
10×10	97.50	84.38	85.57
30×10	97.50	90.00	90.68
30×30	87.50	75.75	76.82
50×10	100.00	92.75	93.41
50×30	95.00	77.13	78.75
50×50	90.00	74.38	75.80
N(0,1)	92.50	81.29	82.31
U[-1,1]	96.67	83.50	84.70
η=0.0	100.00	100.00	100.00
η=0.1	79.17	40.67	44.17
η=0.5	99.17	90.25	91.06
η=1.0	100.00	98.67	98.79
p=2	93.33	76.79	78.30
p=3	95.83	88.00	88.71
Overall	94.58	82.40	83.50

Note: The figures corresponding to columns 'Rational+Random' are computed as weighted means.

From Table 2, we can see that the overall percentages of times in which B-PCA hit global optima was higher than 83% (83.50%). The use of rational starts noticeably decreased the risk of the algorithm to hit local optima. This was observed for all of the levels of the design variables and, of course, during the entire simulation (approximately 94% for the rational start and 82% for the random start). When no noise was added to the data (η=0.0) the B-PCA algorithm always attained the global optima using both rational and random starts. Instead, the B-PCA algorithm seemed to be less prone to hitting global optima when the level of noise was low. More specifically, when η=0.1, we observed the smallest percentages of times in which B-PCA hit global optima (almost 80% for the rational starts and a bit more than 40% for the random starts). As the level of added noise increased, the chance of hitting global optima increased. The other design variables affected the results as follows. The percentages of times in which B-PCA hit global optima increased according to the increase of the number of components *p*. This holds especially when using random starts. The chance of B-PCA to hit global optima seemed to be lower in the *normal* case than in the *uniform* case. Finally, by inspecting Table 2, we can argue that the tendency of hitting local optima increased when *m* increased and the ratio between *n* and *m* decreased. Therefore, when the data matrix was close to be a square matrix and the number of its columns increased, the performance of the B-PCA algorithm seems to get worse. In fact, by ordering (from the highest to the lowest values) the percentages of times in which the global optima was attained with respect to the data sizes, we have: 50×10, 30×10, 10×10, 50×30, 30×30, 50×50. Such an order can be found for the random starts and, to a lesser extent, for the rational starts.

All in all, in order to limit the risk of hitting local optima, more than one starts is recommended. However, the B-PCA algorithm is more prone to hit global optima when rational starts are considered. Moreover, particular attention should be paid when the data set at hand presents a high number of variables, especially if compared with the number of observation units, and a rather low level of noise.

To evaluate the efficiency of the B-PCA algorithms, we compared the computation times both using the rational start and the random starts. It is useful to note that the simulation was carried out on a personal computer with Centrino 1.73 GHz processor and 512 MB RAM. For every level of every design variable, the median computation times are displayed in Table 3. We decided to compute the median values instead of the mean values because the distribution of the computation times was skew and several outliers (very high registered computation times) occurred. To this purpose, we have inspected the boxplots (which are not reported here) for all the combinations of the levels of the design variables (data size, kind of distribution, number of components, level of noise) considering both random and rational starts. We have found that in various conditions there were severe outliers (up to about 1100 seconds). In fact, in 2.58% of runs of the algorithm (in all, 480 data sets × 11 starts = 5280 runs were done), the registered computation time was higher than 60 seconds and in 0.40% of runs higher than 180 seconds. Out of the 96 (combined) conditions in total, in 23 of these we registered at least one computation time higher than 30 seconds. In general, the most time-consuming conditions were those characterized by a high number of observation units (in 17 of the 23 n was 50) and for a low (η=0.1) or medium (η=0.5) level of noise (18 times out of 23). Note also that computation times above 30 seconds were never registered in case η=0.0. Furthermore, it is worth mentioning that in the conditions 10×10, N(0,1), η=1.0, p=2 and 30×10, N(0,1), η=0.1, p=2 and, above all, 50×30, N(0,1), η=0.5, p=3 and 50×50, U(-1,1), η=0.5, p=3, extremely high computation times were registered. Finally, note that, in 57 conditions (out of 96), the registered computation times always remained below 10 seconds.

Table 3. *Efficiency of the B-PCA algorithms: median computation times (seconds) distinguished with respect to the data size, the level of noise, the number of components, the generating distribution and the kind of start.*

Design Variable	B-PCA	
	Rational Start	Random Start
10×10	0.16	0.16
30×10	0.22	0.26
30×30	1.11	1.26
50×10	0.32	0.52
50×30	2.52	3.09
50×50	4.46	4.83
N(0,1)	0.71	0.90
U[-1,1]	0.46	0.54
p=2	0.47	0.57
p=3	0.63	0.76
η=0.0	0.03	0.08
η=0.1	1.62	2.22
η=0.5	1.45	1.88
η=1.0	0.84	0.95
Overall	0.50	0.68

From Table 3, it is interesting to see that the use of rational starts appears to be more helpful. In fact, the increase of the median computation time during the entire simulation when using random starts was equal to about 35% if compared with that resulting from the use of rational starts (the median values are 0.50 for the rational starts and 0.68 for the random starts). Moreover, for all the levels of the design variables, the B-PCA algorithm was more efficient in case of rational starts, except for data size 10×10 in which the difference was negligible. With respect to the design variables, the analysis of the median values showed that, when no noise was added to the data, the algorithm worked very well and converged in about 0.16 seconds (median values). As far as the level of noise added increased, the median computation time firstly increased (when η=0.1) and then decreased, but, however, remained considerably higher than the case η=0.0. Also the number of components affected the computation time, which increased, according to the median values, when *p* passed from 2 to 3. Moreover, the computation time was affected by the kind of distribution. In fact, the algorithm was more efficient for the *uniform* case than the *normal* one. The data size was related to the efficiency of the algorithm as follows. When the number of variables increased, the average computation time increased. The same did not hold for the number of observation units. Nonetheless, in case of equal number of variables, the median computation times increased when the number of observation units increased.

Summing up, similarly to what we found for the tendency of the algorithm to hit global optima, the use of rational starts is recommended for improving the efficiency of B-PCA. In fact, it makes the B-PCA algorithm more efficient when compared to the use of random starts.

5. Application

5.1 Household data

In this subsection, the application of the B-PCA method to the Household data set [13] is described. It is worth to recall that the data at hand contain the percentages of households in *n* = 13 European countries (three countries with missing values were ignored) who consume each of *m* = 20 food items.

Before fitting B-PCA to the data, we performed ordinary B-PCA. In Table 4, we first report the fit percentages and the numbers of unfeasible values obtained performing classical PCA (on the preprocessed data by means of centering and scaling).

Table 4. *Fit percentages from PCA and B-PCA and numbers of unfeasible values from PCA.*

Number of extracted components	Fit Percentage from PCA	Number of unfeasible values from PCA $(\mathbf{AB}' < \underline{\mathbf{X}})$	Number of unfeasible values from PCA $(\mathbf{AB}' > \overline{\mathbf{X}})$	Total number of unfeasible values from PCA	Fit Percentage from B-PCA
$p = 1$	33.54	2	0	2	33.33
$p = 2$	50.22	4	3	7	49.69
$p = 3$	65.19	3	4	7	64.47
$p = 4$	73.80	3	4	7	73.22
$p = 5$	80.38	3	3	6	80.04
$p = 6$	86.05	3	3	6	85.90
$p = 7$	91.17	4	3	7	91.05
$p = 8$	94.98	5	1	6	94.93
$p = 9$	97.37	3	1	4	97.35
$p = 10$	98.76	2	1	3	98.74
$p = 11$	99.83	1	0	1	99.83
$p = 12$	100.00	0	0	0	100.00
$p = 13$	100.00	0	0	0	100.00

From Table 4, we can see that the numbers of unfeasible values remain almost stable passing from *p* = 2 (seven unfeasible values) to *p* = 8 (six unfeasible values). In particular, starting from value *p* = 2, it is interesting to observe that the increase of the number of extracted components does not imply a decrease of the number of unfeasible values (there is an additional unfeasible value when *p* moves from 6 to 7). When *p* = 1, we have a very low number of unfeasible values (only two). We may argue that when the number of extracted components is very low, the component method captures only the most essential structural part of the data. As a consequence, the reconstructed data do not well resemble the anomalous points, i.e. in the current data set those very close to the (lower and upper) bounds. It is surprising that the risk of unfeasible values is not limited to low numbers of extracted components and, indeed, to low fit values. For instance, when *p* = 9, the fit percentage is 97.37% but three unfeasible values occur. Even when *p* = 11, with a fit percentage equal to 99.83%, we get one reconstructed datum lower than 0%.

Because there were unfeasible values for all relevant numbers of components, it seems useful to apply B-PCA to these data in order to avoid unfeasible values. In order to determine the optimal number of components, we can inspect the last column of Table 4, where the fit percentages obtained performing B-PCA with *p* = 1,…, 13. are given. We chose *p* = 3 components. This is because passing from two to three components implies a noticeably increase of the fit percentage. Moreover, using three components guarantees that all the foods (except Butter and, to a lesser extent, Olive Oil) are well captured by at least one component as one can see in Table 5, which contains the varimax rotated component loadings. For the sake of completeness, we also reported the loadings from PCA (within parentheses). Note that the given PCA loadings were rotated so that they resembled as much as possible the varimax rotated B-PCA loadings.

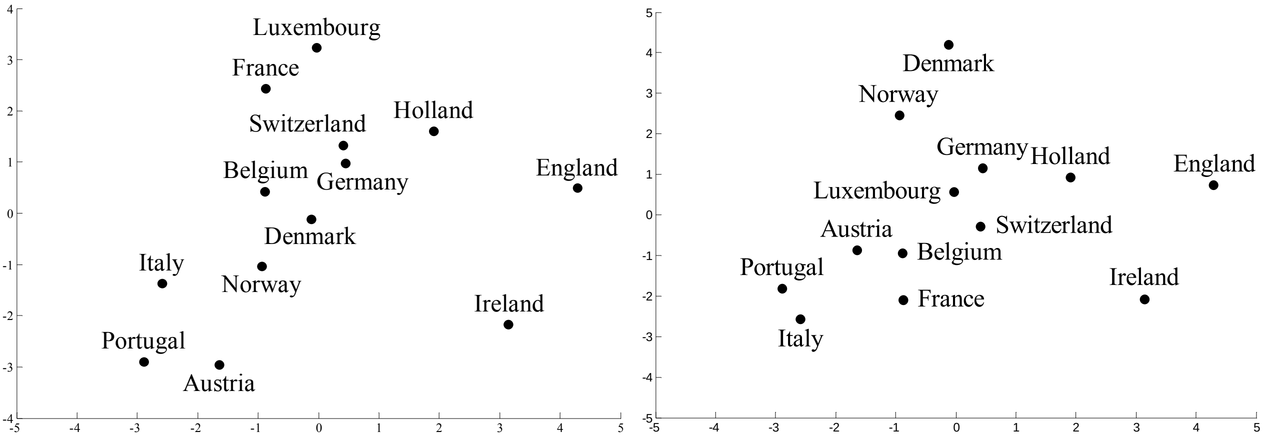
Table 5. *Varimax rotated component loadings from B-PCA and (within parentheses) from PCA.*

Food	Component 1	Component 2	Component 3
Ground Coffee	**-0.38** (**-0.36**)	**0.28** (**0.31**)	0.19 (0.17)
Instant Coffee	**0.35** (**0.33**)	0.20 (0.19)	-0.14 (-0.12)
Tea	**0.25** (**0.30**)	-0.05 (-0.06)	0.19 (0.17)
Sweetener	0.13 (0.12)	0.12 (0.13)	**0.33** (**0.31**)
Biscuits	**0.25** (**0.24**)	0.17 (0.17)	-0.02 (-0.01)
Powder Soup	**0.28** (**0.26**)	**0.21** (**0.21**)	-0.15 (-0.16)
Tinned Soup	**0.32** (**0.36**)	-0.02 (0.03)	-0.00 (0.04)
Industrial Potatoes	-0.01 (0.01)	**0.24** (**0.23**)	0.11 (0.10)
Frozen Fish	-0.12 (-0.12)	-0.01 (-0.00)	**0.49** (**0.47**)
Frozen Vegetables	0.03 (0.02)	0.07 (0.07)	**0.43** (**0.42**)
Apples	0.07 (0.08)	**0.40** (**0.40**)	0.06 (0.05)
Oranges	-0.08 (-0.07)	**0.47** (**0.47**)	0.05 (0.05)
Tinned Fruit	**0.29** (**0.27**)	**0.24** (**0.24**)	0.09 (0.09)
Jam	**0.42** (**0.40**)	-0.10 (-0.10)	0.04 (0.02)
Garlic	**-0.28** (**-0.29**)	**0.23** (**0.23**)	**-0.26** (**-0.24**)
Butter	0.12 (0.11)	0.13 (0.13)	-0.05 (-0.04)
Margarine	-0.02 (-0.05)	0.08 (0.05)	**0.26** (**0.37**)
Olive Oil	0.18 (-0.17)	0.12 (0.11)	-0.17 (-0.15)
Yoghurt	-0.03 (-0.02)	**0.43** (**0.43**)	-0.12 (-0.13)
Crisp Bread	-0.03 (-0.03)	-0.06 (-0.06)	**0.39** (**0.36**)

Note: Loadings higher than 0.20 in the absolute sense are in bold.

By inspecting Table 5, we can observe that the component loadings from PCA and B-PCA slightly differ. It follows that the interpretation of the components of the two solutions is the same for all practical purposes. In particular, the first component can be interpreted as the style of nutrition characterized by rich and quick breakfast. For instance, Instant Coffee is highly partaken, whereas Ground Coffee is often missing because its preparation requires more time). The second component seems to be representative of healthy and quickly preparable food. Finally, the third component mainly reflects the presence of dietary food (especially, Frozen Fish and Frozen Vegetables, Crisp Bread and Sweetener). The B-PCA component scores for the Countries are displayed in Figure 1.

Figure 1. *Low dimensional representation from B-PCA (Left side: Components 1 and 2; Right side: Components 2 and 3)*



Note: *x*-axes: Component 1; *y*-axes: Component 2

Note: *x*-axes: Component 1; *y*-axes: Component 3

Starting from the right side of Figure 1, in which the scores of the third component are represented, we can observe the duality between the countries belonging to Northern Europe (Denmark and Norway above all) with high third component scores and those located in Southern Europe (Italy, France and Portugal) with low third component scores, even if the exception of Ireland is visible. We may argue that in Southern Europe the consumption of frozen foods is lower than that in the Northern countries, whereas the opposite comment holds for garlic. Thus, such a component also reflects the geographical location and the culinary culture of the countries. The highest first component scores correspond to two countries located in the United Kingdom, which are in contrast with the countries from Southern Europe (characterized by the lowest first component scores). Finally, by inspecting the second component scores, we can state that this component reflects, to some extent, the distinction between the wealthy European countries (among them Luxembourg, France, Holland and Switzerland) and the other ones (Italy, Ireland, and Portugal, even if Austria also appears).

To sum up, we conclude that ordinary PCA gives unfeasible values in solutions with all reasonable numbers of components. B-PCA can be used to avoid this problem, without losing the interpretability of the ordinary PCA solution.

5.2 Greek red wine data

The data here considered refer to the Anthocyanin concentrations of Greek red wines from the 1998 vintage. The data can be found in Table 4 of Reference [18]. Specifically, n = 19 wines were tested with respect to the presence of m = 6 Anthocyanin concentrations. In this case, the constraints concerned the lower bounds of the data. In fact, it was desirable that the estimated chemical concentrations, expressed as mg/l, took non-negative values, whereas no attention was paid to the upper bound constraints. Note that a peculiarity of the data set at hand is the presence of several scores equal to 0 (28.9% of scores) for which the chance of getting negative estimated values by using ordinary PCA is high.

The data were preprocessed (by centering and scaling) and PCA and B-PCA were then applied. The presence of unfeasible values resulting from PCA and the fit percentages of PCA and B-PCA are given in Table 6. Note that the lowest unfeasible values (from PCA) after backtransforming the data are displayed.

Table 6. *Fit percentages from PCA and B-PCA and numbers of unfeasible values from PCA.*

Number of extracted components	Fit Percentage from PCA	Number of unfeasible values from PCA ($\mathbf{AB}' < \underline{\mathbf{X}}$)	Lowest unfeasible value from PCA (mg/l)	Fit Percentage from B-PCA
$p = 1$	63.44	8	-25.1	63.25
$p = 2$	83.00	15	-443.1	81.08
$p = 3$	95.57	21	-80.0	95.21
$p = 4$	98.75	13	-8.1	98.55
$p = 5$	99.79	16	-3.1	99.71
$p = 6$	100.00	0	-	100.00

From Table 6, we can observe that, apart from the trivial case with $p = 6$, the number of unfeasible values obtained performing ordinary PCA ranges from 8 (7.0% of the estimated data) for $p = 1$ to 21 (18.4% of the estimated data) for $p = 3$. The registered number of negative estimated values does not decrease according to the increase of the number of extracted components. In fact, also when $p = 5$, 14.0% of data are estimated by values lower than 0. Finally, it is surprising to see the notably unlucky estimated value (applying the inverse preprocessing procedure) regarding the concentration of Acylated malvidin-3-glucoside for wine 10, which is equal to -443.1 mg/l considering $p = 2$ components (the observed score is 25.8 mg/l). On the basis of these results, we can conclude that the use of B-PCA is highly advisable here.

A reasonable choice seemed to be the use of B-PCA with $p = 3$ components. The obtained varimax rotated component loadings are displayed in Table 7 (the PCA loadings are within parentheses and were rotated so as to become as similar as possible to the B-PCA loadings).

Table 7. *Varimax rotated component loadings from B-PCA and (within parentheses) from PCA.*

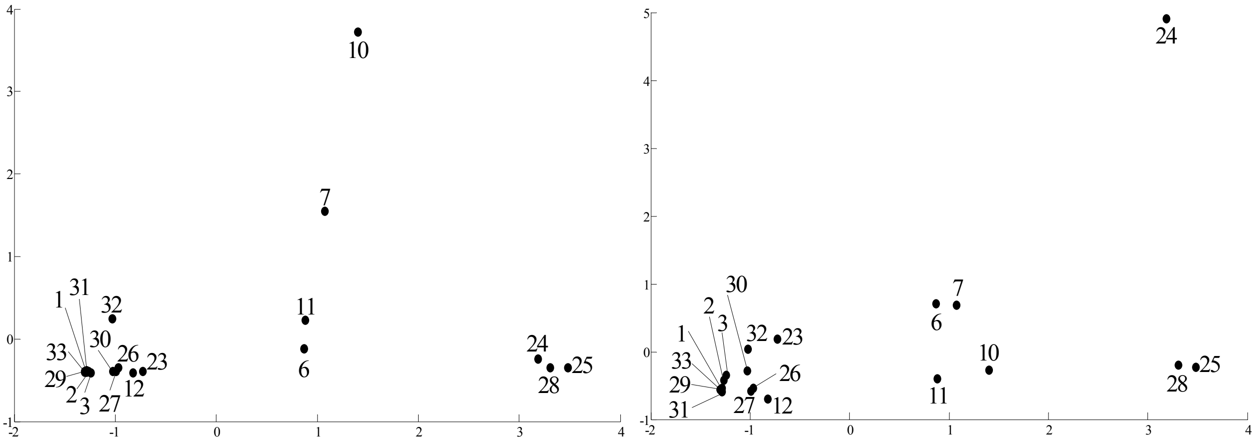
Anthocyanin concentrations	Component 1	Component 2	Component 3
Delphinidin-3-glucoside	**0.66** (**0.66**)	-0.10 (-0.09)	**-0.37** (**-0.37**)
Cyanidin-3-glucoside	-0.01 (0.01)	**0.99** (**0.99**)	-0.02 (-0.02)
Petunidin-glucoside	**0.45** (**0.43**)	0.07 (0.06)	0.23 (0.23)
Peonidin-3-glucoside	**0.38 (0.39)**	-0.02 (-0.02)	**0.35** (**0.34**)
Malvidin-3-glucoside	**0.47** (**0.47**)	0.10 (0.10)	0.14 (0.14)
Acylated-malvidin-3-glucoside	-0.07 (-0.07)	-0.05 (-0.05)	**0.82** (**0.82**)

Note: Loadings higher than 0.30 in the absolute sense are in bold.

The second component is essentially influenced by exactly one variable (Cyanidin-3-glucoside). The first and third components are mainly related to the remaining Anthocyanin concentrations. Specifically, Petunidin-glucoside and Malvidin-3-glucoside strongly affect the first component, Acylated-malvidin-3-glucoside the third one. Finally, Delphinidin-3-glucoside and Peonidin-3-glucoside are related to both Components 1 and 3.

The low-dimensional configuration of the wines resulting from B-PCA is represented in Figure 2.

Figure 2. *Low dimensional representation from B-PCA (Left side: Components 1 and 2; Right side: Components 2 and 3)*



Note: *x*-axes: Component 1; *y*-axes: Component 2

Note: *x*-axes: Component 1; *y*-axes: Component 3

In Kallithraka et al. (2001) the variety and the geographical origin of the wines are also given as external information. The low-dimensional plots is not able to clearly distinguish the wines with respect to their variety. With geographical origin, however, there is a clear relation: the wines from Southern Greece (24, 25 and 28) are easily distinguished from the others. In fact, these are characterized by the highest first component scores. All the wines from the Greek islands (29-33) have low first component scores.

All in all, differently from PCA, once more the use of the B-PCA method provides a feasible low-dimensional approximation of the data set without affecting the interpretability of the obtained solution if compared with that resulting from PCA.

6. Conclusion

In this paper, a constrained generalization of PCA, called Bounded Principal Component Analysis (B-PCA) has been proposed. The need for B-PCA arises when the observed data are lower and/or upper bounded. In fact, a possible limit resulting from performing classical PCA is the risk of extracting meaningless components in the sense that the reconstructed data are out of the observed bounds. In order to determine the optimal component matrices from B-PCA a constrained (row-wise) ALS algorithm has been proposed. This is based on the iterative solution of ad-hoc LSI problems (Lawson, Hanson, 1995). A comparison between B-PCA and ordinary PCA in recovering the underlying structure of the data has been done and the performance of the B-PCA algorithm in terms of the frequency of hitting global optima and of the computation time has been tested by means of a simulation experiment. The results are satisfactory and encourage new lines of research. Among them, it will be interesting to deal with three-way bounded data by suitably developing constrained generalizations of the Tucker3 [19] and Candecomp/Parafac [20-21] models. These could be found as straightforward generalizations of (8) in which $\mathbf{B}$ is replaced by $(\mathbf{C}\otimes\mathbf{B})\mathbf{G}_a'$ for the Tucker3 model (symbol $\otimes$ denotes the Kronecker product) and $(\mathbf{C}\odot\mathbf{B})$ for the Candecomp/Parafac model (symbol $\odot$ denotes the Khatri-Rao product). Here $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{C}$ denote the component matrices for the observation unit, variable and occasion modes and, in Tucker3, $\mathbf{G}_a$ is the observation unit mode matricized version of the so-called core matrix (see, for more details, Tucker, 1966). The optimal solution could be found by iteratively solving LSI algorithms with respect to the rows of the component matrices for the observation unit, variable and occasion modes and, in the

Tucker3 case, of the core matrix. Unfortunately, prior analyses of such a three-way B-PCA algorithm showed a very strong tendency of the algorithm to hit local optima. It is clear that specific tools for managing this drawback should be found.

References

[1] Carroll, J.D., Pruzansky, S, and Kruskal, J.B., CANDELINC: A general approach to multidimensional analysis of many-way arrays with linear constraints on parameters, *Psychometrika*, **45** (1980) 3-24.

[2] van der Kloot, W.A. and Kroonenberg, P.M., External analysis with three-mode principal component models, *Psychometrika*, **50** (1985) 479-494.

[3] Takane, Y., Kiers, H.A.L. and de Leeuw, J., Component analysis with different sets of constraints on different dimensions, *Psychometrika*, **60** (1995) 259-280.

[4] Bro, R., and de Jong, S., A fast non negativity-constrained least squares algorithm, *Journal of Chemometrics*, **11** (1997) 393-401.

[5] de Juan, A., Vander Heyden, Y., Tauler, R., and Massart, D.L., Assessment of new constraints applied to the alternating least squares method, *Analytica Chimica Acta*, **346** (1997) 307-318.

[6] Bro, R., *Multi-way Analysis in the Food Industry. Models, Algorithms, and Applications*. Ph.D. thesis, University of Amsterdam & Royal Veterinary and Agricultural University, 1998.

[7] Bro, R., and Sidiropoulos, N.D., Least squares algorithms under unimodality and non-negativity constraints, *Journal of Chemometrics*, **12** (1998) 223-247.

[8] Kiers, H.A.L. and Smilde, A.K. Constrained three-mode factor analysis as a tool for parameter estimation with second-order instrumental data, *Journal of Chemometrics*, **12** (1998) 125-147.

[9] Takane, Y., and Hunter, M. A., Constrained principal component analysis: a comprehensive theory, *AAECC*, **12** (2001) 391-419.

[10] Bijlsma, S., Boelens, H.F.M., Hoefsloot, H.C.J., and Smilde, A.K., Constrained least squares methods for estimating reaction rate constants from spectroscopic data, *Journal of Chemometrics*, **16** (2002) 28-40.

[11] Van Benthem, M .H., Keenan, M., and Haaland, D., Application of equality constraints on variables during alternating least squares procedures, *Journal of Chemometrics*, **16** (2002) 613-622.

[12] van Benthem, M.H., and Keenan, M.R., Fast algorithm for the solution of large-scale non-negativity-constrained least squares problems, *Journal of Chemometrics*, **18** (2004) 441-450.

[13] Trygg, J., Chemometrics made easy - understanding food consumption patterns in Europe with principal component analysis (PCA), *Homepage of Chemometrics*, (2003) (http://www.acc.umu.se/~tnkjtg/Chemometrics/Editorial).

[14] Lawson, C.L., and Hanson, R.J., *Solving Least Squares Problems*. Classics in Applied Mathematics, vol. 15, SIAM, Philadelphia, PA (1995).

[15] van Perlo-ten Kleij, F., *Contributions to Multivariate Analysis with Applications in Marketing*, Ph.D. thesis, University of Groningen, 2004.

[16] Cutler, A., and Breiman, L., Archetypal analysis, *Technometrics*, **36** (1994) 338–347.

[17] Timmerman, M.E., and Kiers, H.A.L., Three-mode principal component analysis: Choosing the number of components and sensitivity to local optima, *British Journal of Mathematical and Statistical Psychology*, **53** (2000) 1-16.

[18] Kallithraka, S., Arvanitoyannis, I.S., Kefalas, P., El-Zajouli, A., Soufleros, E., and Psarra, E., Instrumental and sensory analysis of Greek wines; implementation of principal

component analysis (PCA) for classification according to geographical origin, *Food Chemistry*, **73** (2001) 501-514.

[19] Tucker, L.R, Some mathematical notes on three-mode factor analysis, *Psychometrika*, **31** (1966) 279-311.

[20] Carroll, J.D., and Chang, J.J., Analysis of individual differences in multidimensional scaling via an *n*-way generalization of Eckart-Young decomposition, *Psychometrika*, **35** (1970) 283-319.

[21] Harshman, R.A., Foundations of the PARAFAC procedure: models and conditions for an "explanatory" multi-mode factor analysis, *UCLA Working Papers in Phonetics*, **16** (1970) 1-84.